IBM Systems - iSeries

Archivos y sistemas de archivos
Sistema de archivos integrado

Versión 5 Release 4

IBM Systems - iSeries

Archivos y sistemas de archivos
Sistema de archivos integrado

Versión 5 Release 4

Nota

Antes de utilizar esta información y el producto al que hace referencia, lea la información que figura en: “Notas”, en la página 149.

Séptima edición (febrero de 2006)

Esta edición corresponde a la versión 5, release 4, modificación 0 de IBM i5/OS (producto número 5722–SS1) y a todos los releases y modificaciones posteriores a menos que se indique lo contrario en nuevas ediciones. Esta versión no se ejecuta en todos los modelos RISC (Reduced Instruction Set Computer), ni tampoco se ejecuta en los modelos CICS.

Contenido

Sistema de archivos integrado

El sistema de archivos integrado es un componente de i5/OS que admite la gestión de almacenamiento y entrada/salida continua similar a los sistemas operativos de PC y UNIX, al mismo tiempo que proporciona una estructura integrada para toda la información almacenada en el servidor.

Nota: Al utilizar los ejemplos de código, acepta los términos de la "Información de licencia de código y declaración de limitación de responsabilidad" en la página 148.

Novedades de V5R4

En este tema se describen los cambios realizados en este temario de V5R4.

QNTC

QNTC da ahora soporte al puerto TCP/IP 445 y a tamaños de archivos mayores.

- QNTC da soporte al puerto TCP/IP 445

  El sistema de archivos QNTC ahora puede contactar con servidores utilizando el puerto TCP/IP 445. No es necesario que los servidores Windows estén configurados para NetBios sobre TCP/IP.
- QNTC da soporte a tamaños de archivos mayores

  A partir de V5R4, el sistema de archivos QNTC da soporte a la lectura y escritura de archivos de hasta 1 TB (1 TB es aproximadamente igual a 1.099.511.627.776 de bytes).

Mandato Reclamar enlaces de objetos (RCLLNK)

El mandato Reclamar enlaces de objetos (RCLLNK) identifica y repara los objetos dañados en los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario montados sin necesidad de que el sistema esté en un estado restringido. Esto permite corregir los problemas en estos sistemas de archivos sin disminuir la productividad. En muchos casos, se puede utilizar como una alternativa al mandato Reclamar almacenamiento (RCLSTG). Por ejemplo, RCLLNK está especialmente indicado para identificar y corregir problemas en los siguientes casos:

- Los problemas están aislados en un único objeto.
- Los problemas están aislados en un grupo de objetos.
- Los objetos dañados se tienen que identificar o suprimir.
- El sistema no puede estar en un estado restringido durante la operación de reclamación.
- Las agrupaciones de almacenamiento auxiliar (ASP) independientes deben estar disponibles durante la operación de reclamación.

Cómo ver las novedades o las modificaciones

Para ayudarle a ver dónde se han realizado cambios técnicos, en este documento se utiliza:

- La imagen » que marca dónde empieza la información nueva o modificada.
- La imagen « que marca dónde termina la información nueva o modificada.

Para obtener más información sobre las novedades o las modificaciones en este release, consulte el Memorándum de usuarios.

Archivo PDF imprimible

Este tema le indicará cómo ver e imprimir un archivo PDF de esta información.

Para ver o descargar la versión PDF de este documento, seleccione Sistemas de archivos integrados (aproximadamente, 1845 KB).

Cómo guardar los archivos PDF

Si desea guardar un archivo PDF en su estación de trabajo para verlo o imprimirlo:

1. Pulse el PDF con el botón derecho del ratón en el navegador (pulse el enlace anterior con el botón derecho del ratón).
2. Pulse la opción que guarda el PDF localmente.
3. Navegue hasta el directorio en el que desea guardar el PDF.
4. Pulse **Guardar**.

Cómo descargar Adobe Reader

Para poder ver o imprimir archivos PDF, debe instalar Adobe Reader en su sistema. Puede descargar una copia gratuita desde el sitio Web de Adobe (www.adobe.com/products/acrobat/readstep.html) .

Visión general del sistema de archivos integrado

A continuación se proporciona información sobre sistema de archivos integrado en el servidor iSeries y la utilidad que puede tener en el servidor.

¿Qué es el sistema de archivos integrado?

El *sistema de archivos integrado* es un componente de i5/OS 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.

El sistema de archivos integrado engloba 11 sistemas de archivos, con sus propios conjuntos de estructuras y reglas lógicas para interaccionar con la información situada en el almacenamiento.

Las características principales 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 solo 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 para iSeries o en un servidor Windows NT remoto. Los archivos continuos también pueden almacenarse de forma 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).

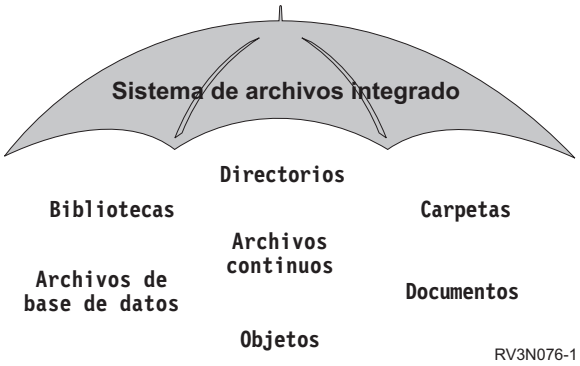

Figura 1. Una estructura que abarca toda la información almacenada en el servidor iSeries

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

¿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 de i5/OS 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.

Puede utilizar el sistema de archivos integrado para lo siguiente:

- Ofrecer un acceso más rápido a los datos de i5/OS, especialmente para aplicaciones como iSeries Access que utilizan el servidor de archivos i5/OS.
- 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 el sistema operativo UNIX como, por ejemplo, POSIX (Interfaz portátil de sistemas operativos para entornos informáticos) y XPG (X/Open Portability Guide). 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 Windows.
- Permitir manejar el soporte de archivos con posibilidades exclusivas (como los archivos de base de datos orientados a registros, archivos continuos basados en el sistema operativo UNIX y servicio de archivos) como sistemas de archivos independientes, al mismo tiempo que permite manejarlos mediante una interfaz común.

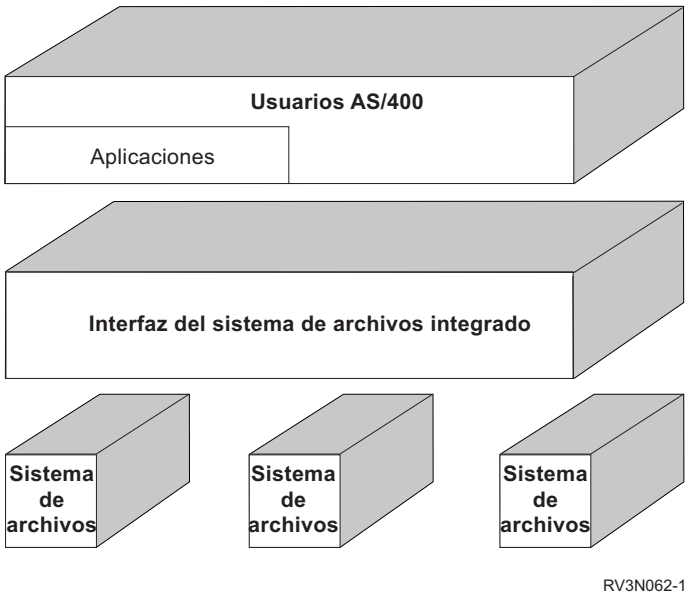

Figura 2. Una interfaz común para distintos sistemas de archivos

- 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 del servidor iSeries y otros objetos del mismo modo 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.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

Conceptos del sistema de archivos integrado

En este tema se hace una introducción a los conceptos básicos del sistema de archivos integrado como, por ejemplo, el directorio, el enlace, el nombre de vía de acceso, el archivo continuo, la continuidad de nombres, los atributos ampliados y el soporte de exploración.

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 los subárboles de los sistemas de archivos QSYS.LIB y QSYS.LIB de ASP independiente, 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 la posibilidad de almacenar un archivo una sola vez pero acceder al mismo a través de múltiples vía s de acceso utilizando enlaces.

Los sistemas de archivos y los objetos son ramas del árbol de directorios del sistema de archivos integrado. En la figura siguiente se muestra un ejemplo de un árbol de directorios del sistema de archivos integrado.

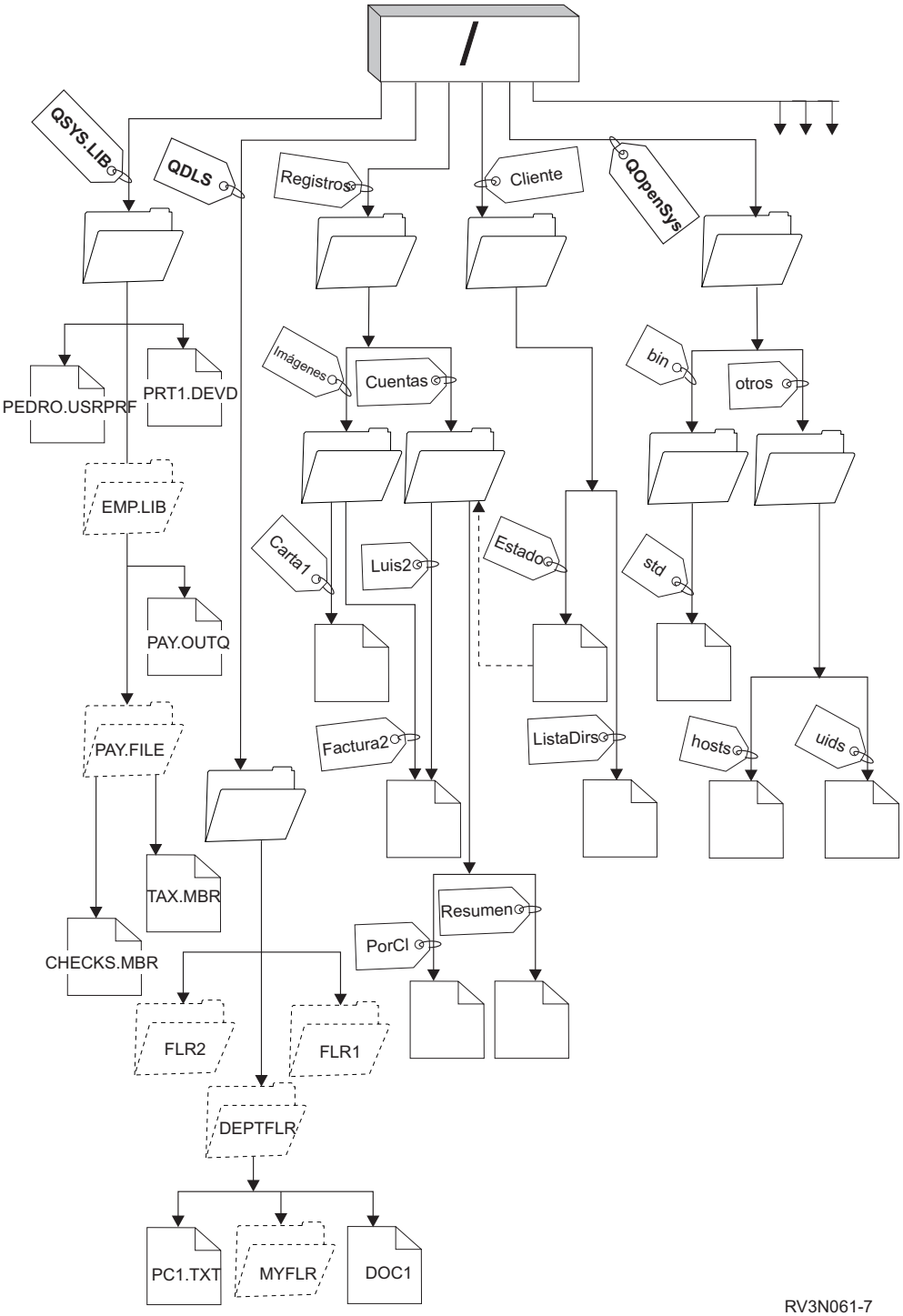

Figura 3. Ejemplo de árbol de directorios del sistema de archivos integrado

Directorio actual

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 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.

Directorio inicial

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 preferible crear directorios iniciales individuales para cada usuario bajo el directorio /home. El directorio /home es un subdirectorio 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.

Puede especificarse 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.

Conceptos relacionados

"Directorios proporcionados"
El sistema de archivos integrado crea estos directorios cuando se reinicia el sistema, si todavía no existen.

Información relacionada

Mandato Cambiar directorio actual (CHGCURDIR)

chdir()

fchdir()

Interfaces de programación de aplicaciones (API)

Directorios proporcionados

El sistema de archivos integrado crea estos directorios cuando se reinicia el sistema, si todavía no existen.

Nota: No sustituya los siguientes directorios creados por el sistema por enlaces simbólicos con otros objetos. Por ejemplo, no sustituya /home por un enlace simbólico a un directorio en una ASP independiente. De lo contrario, se pueden producir problemas en la ASP independiente o al crear nuevos perfiles de usuario.

/tmp El directorio /tmp proporciona a las aplicaciones un lugar donde almacenar objetos 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 objeto en el directorio /tmp, el objeto permanece allí hasta que la aplicación o el usuario lo eliminan. El sistema no elimina automáticamente objetos del directorio /tmp ni ejecuta ningún otro proceso especial para los objetos que se encuentran en el directorio /tmp.

Puede utilizar los mandatos y pantallas de usuario que dan soporte al sistema de archivos integrado para gestionar el directorio /tmp y sus objetos. Por ejemplo, puede utilizar la pantalla Trabajar con enlaces de objetos o el mandato WRKLNK para copiar, eliminar o redenominar el directorio /tmp o los objetos 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 dan soporte al sistema de archivos integrado para gestionar el directorio /tmp y sus objetos. Por ejemplo, el programa de aplicación puede eliminar un objeto del directorio /tmp utilizando la API unlink().

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

El directorio /tmp puede tener el atributo restricted rename and unlink establecido en Yes a efectos de seguridad y elementos comunes del sistema operativo.

Nota: El atributo restricted rename and unlink es equivalente al bit de modalidad S_ISVTX de un directorio.

Si el atributo restricted rename and unlink se establece en Yes, no puede redenominar o desenlazar objetos dentro del directorio /tmp, a menos que se cumpla una de las siguientes condiciones:

- Es el propietario del objeto.
- Es el propietario del directorio.
- Tiene autoridad especial (*ALLOBJ) para todos los objetos.

Si el atributo se establece en Yes y no tiene las autoridades correspondientes, aparecerá el error número 3027 (EPERM) o el mensaje MSGCPFA0B1 (La operación solicitada no está permitida. Problema de acceso) para las anomalías de redenominación o desenlace cuando se utilizan los siguientes mandatos y API:

- Mandato Eliminar enlace (RMVLNK, DEL y ERASE)
- Mandato Eliminar directorio (RMVDIR, RD y RMDIR)
- Mandato Redenominar objeto (RNM y REN)
- Mandato Mover objeto (MOV y MOVE)
- API Redenominar archivo o directorio (rename())
- API Redenominar archivo o directorio, mantener "nuevo" si existe (Qp0lRenameKeep())
- API Redenominar archivo o directorio, desenlazar "nuevo" si existe (Qp0lRenameUnlink())
- API Eliminar directorio (rmdir())
- API Eliminar enlace con archivo (unlink())

El atributo restricted rename and unlink y el bit de modalidad S_ISVTX se pueden modificar utilizando el mandato Cambiar atributo (CHGATR) o las API Establecer atributos (Qp0lSetAttr())

o Cambiar autorizaciones de archivo (chmod) si es el propietario del objeto, o si tiene la autoridad especial (*ALLOBJ) para todos los objetos. No obstante, si el atributo se cambia a No, perderá las ventajas de seguridad y elementos comunes del sistema operativo que proporciona el valor Yes.

| Cuando se crea el directorio /tmp durante el reinicio del sistema, el atributo se establece en Yes. Si
| el directorio /tmp ya existe durante el reinicio del sistema, el atributo no se cambia.

/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.

/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 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 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.

/dev El directorio /dev contiene varios directorios y archivos del sistema.

/dev/xti
El directorio /dev/xti contiene los controladores de dispositivo UDP Y TCP.

Conceptos relacionados

"Directorio inicial" en la página 7
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.

Referencia relacionada

"Dispositivos UDP y TCP en el sistema de archivos ″raíz″ (/)" en la página 35
El sistema de archivos "raíz" (/) bajo el directorio /dev/xti mantendrá ahora dos controladores de dispositivos denominados udp y tcp.

"Sistema de archivos de sistemas abiertos (QOpenSys)" en la página 36
El sistema de archivos QOpenSys es compatible con los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX y X/Open Portability Guide (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.

"Recrear objetos proporcionados por el sistema de archivos integrado" en la página 111
Esta tabla muestra los objetos proporcionados por el sistema de archivos integrado que vuelve a crear el mandato Reclamar enlaces de objetos (RCLLNK) si no existen. Estos objetos se crean normalmente durante la carga del programa inicial (IPL). Si es necesario, también puede recrear algunos de estos objetos utilizando el mandato Reclamar almacenamiento (RCLSTG).

Información relacionada

Mandato WRKLNK

Directorios *TYPE2

Los sistemas de archivos ″raíz″ (/), QOpenSys y 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.

Nota: El concepto de los archivos continuos *TYPE1 y *TYPE2 es distinto del concepto de los formatos de directorios *TYPE1 y *TYPE2. No están relacionados.

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
- En muchos casos, menos espacio de almacenamiento auxiliar

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:

- Los directorios *TYPE2 permiten cambiar el nombre de mayúsculas a minúsculas y viceversa en un sistema de archivos que admita solo mayúsculas o solo minúsculas (por ejemplo, pasar de B a b).
- Un objeto de un directorio *TYPE2 puede tener un máximo de un millón de enlaces, mientras que los directorios *TYPE1 solo 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 999.998 subdirectorios.
- Con el iSeries Navigator, la lista de entradas se clasifican automáticamente en orden binario al abrir un directorio con el formato *TYPE2.
- Existen funciones nuevas, como el soporte de la exploración del sistema de archivos integrado, que solo están disponibles para los objetos de los directorios *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:

- Los nuevos servidores de iSeries en los que previamente se ha instalado OS/400 V5R2 o i5/OS V5R3 o posterior tienen directorios *TYPE2. No es necesario ningún tipo de conversión para los sistemas de archivos ″raíz″ (/), QOpenSys y UDFS en las ASP 1 a 32.
- Una instalación de OS/400 V5R2 o i5/OS V5R3 o posterior partiendo de cero en un servidor iSeries incluye directorios *TYPE2. No es necesario ningún tipo de conversión para los sistemas de archivos ″raíz″ (/), QOpenSys y UDFS en las ASP 1 a 32.
- La utilidad de conversión de la V5R1 o V5R2 se utiliza para convertir los sistemas de archivos.
- Si los UDFS en una ASP todavía no se han convertido al formato *TYPE2, se convertirán la primera vez que se active la ASP independiente en un sistema instalado con OS/400 V5R2 o i5/OS V5R3 o posterior.
- El sistema convierte automáticamente el resto de sistemas de archivos admitidos excepto UDFS en ASP independientes que sigan utilizando directorios *TYPE1. Esta conversión empieza después de la instalación de i5/OS V5R3M0 o releases posteriores. No debería afectar demasiado a la actividad del sistema.

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

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

Nota: En OS/400 V5R2 o i5/OS V5R3, se admiten los directorios *TYPE2, pero existen algunas diferencias con el soporte de directorios *TYPE2 normal.

Utilización de los directorios *TYPE2 en OS/400 V5R1 o V5R2

Los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario (UDFS) en el sistema de archivos integrado admiten el formato de los directorios *TYPE2 en OS/400 V5R1, V5R2 y posteriores.

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 OS/400 V5R1 o V5R2, puede convertir sus directorios al formato de directorios *TYPE2 utilizando la utilidad de conversión apropiada. Poco después de la instalación de i5/OS V5R3M0 o un release posterior, el sistema empezará a convertir automáticamente a directorios *TYPE2 cualquier sistema de archivos que todavía no de soporte a los directorios *TYPE2. Por lo tanto, quizás desee realizar una conversión al formato de directorios *TYPE2 antes de instalar una versión posterior para evitar esta conversión automática.

El soporte para directorios *TYPE2 en OS/400 V5R2 está disponible en V5R2 iSeries Information Center mediante el mandato Convertir directorio (CVTDIR).

Los arreglos temporales del programa (PTF) permiten obtener soporte para los directorios *TYPE2 en OS/400 V5R1. La herramienta de conversión es algo diferente de la utilizada en OS/400 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:

- Descargue el APAR informativo en su servidor iSeries y visualícelo. Utilice los mandatos siguientes:

```
SNDPTFORD PTFID((II13161))
DSPPTFCVR LICPGM(INFOAS4) SELECT(II13161)
```

- Vaya a www.ibm.com/eserver/iseries/support/supporthome.nsf/document/10000045 para ver el APAR informativo. Consulte **Problem Solving → Technical Databases → Authorized Program Analysis Reports (APARs) → V5R1 APARs → APAR number II13161**.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

"Visión general de la conversión de *TYPE1 a *TYPE2" en la página 97
Los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

Referencia relacionada

"Conversión de directorios de *TYPE1 a *TYPE2" en la página 96
Los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

"Nombres de vía de acceso en el sistema de archivos ″raíz″ (/)" en la página 34
Los nombres de vías de acceso tiene un formato específico en el sistema de archivos "raíz" (/).

"Nombres de vía de acceso en un sistema de archivos definido por usuario del sistema de archivos integrado" en la página 40
Un objeto archivo especial de bloqueo (*BLKSF) representa un sistema de archivos definido por el usuario (UDFS) cuando hay que manipular la totalidad del UDFS y todos los objetos que hay en él.

Información relacionada

Mandato Reclamar almacenamiento (RCLSTG)

Mandato Convertir directorio (CVTDIR)

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 directorio la que identifica el 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 "Directorio inicial" en la página 7). Determinados tipos de enlace pueden cruzar sistemas de archivos y pueden existir sin que exista un objeto.

Existen dos tipos de enlaces: 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. 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 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.

Referencia relacionada

"Enlaces en el sistema de archivos ″raíz″ (/)" en la página 34
Se permiten varios enlaces fijos con el mismo objeto del sistema de archivos "raíz" (/). Los enlaces simbólicos se soportan totalmente.

"Enlaces en el sistema de archivos QOpenSys" en la página 37
Se permiten múltiples enlaces fijos con el mismo objeto del sistema de archivos QOpenSys. Los enlaces simbólicos se soportan totalmente.

"Enlaces en un sistema de archivos definido por usuario del sistema de archivos integrado" en la página 41
Un sistema de archivos definido por el usuario (UDFS) permite múltiples enlaces fijos con el mismo objeto y soporta plenamente los enlaces simbólicos.

"Enlaces en el sistema de archivos QSYS.LIB" en la página 47
No pueden crearse o almacenarse enlaces simbólicos en el sistema de archivos QSYS.LIB.

"Enlaces en el sistema de archivos QSYS.LIB de ASP independiente" en la página 51
No pueden crearse o almacenarse enlaces simbólicos en el sistema de archivos QSYS.LIB de ASP independiente.

"Enlaces en el sistema de archivos QDLS" en la página 53
Los enlaces simbólicos no se pueden crear ni almacenar en el sistema de archivos QDLS.

"Enlaces en el sistema de archivos QOPT" en la página 56
El sistema de archivos QOPT da soporte solo a un enlace con un objeto. Los enlaces simbólicos no se pueden crear ni almacenar en QOPT.

"Enlaces en el sistema de archivos QNTC" en la página 62
El sistema de archivos QNTC da soporte solo a un enlace con un objeto. No puede crear o almacenar enlaces simbólicos en QNTC.

"Enlaces en el sistema de archivos QFileSvr.400" en la página 67
El sistema de archivos QFileSvr.400 da soporte solo a un enlace con un objeto.

"Enlaces en el sistema de archivos de red" en la página 70
En general, en el sistema de archivos de red se permiten varios enlaces fijos con el mismo objeto.

"Consejos: enlace simbólico" en la página 100
Los enlaces simbólicos son objetos del sistema de archivos integrado que contienen una vía de acceso a otro objeto.

"Enlaces en el sistema de archivos QNetWare" en la página 59
El sistema de archivos QNetWare da soporte solo a un enlace con un objeto. En QNetWare no pueden crearse ni almacenarse enlaces simbólicos.

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. Solo puede haber un enlace fijo de un directorio con otro directorio.

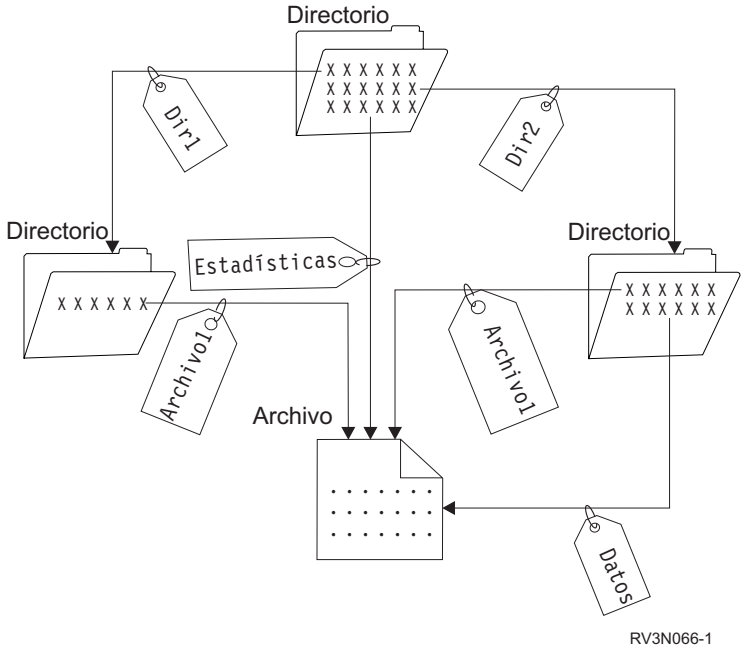

Figura 4. Una entrada de directorio define cada enlace fijo

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 a los sistemas de archivos QSYS.LIB o QSYS.LIB de ASP independiente y al sistema de archivos de servicios de biblioteca de documentos (QDLS), aunque 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 con el *mismo objeto* en los sistemas de archivos QSYS.LIB, QSYS.LIB de ASP independiente o 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.

Conceptos relacionados

"Consideraciones para múltiples enlaces fijos y el registro por diario" en la página 107
Si posee múltiples enlaces fijos a un objeto del sistema de archivos integrado registrado por diario, todos los enlaces deben salvarse y restaurarse conjuntamente de forma que se mantenga el enlace y la información asociada del diario.

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:

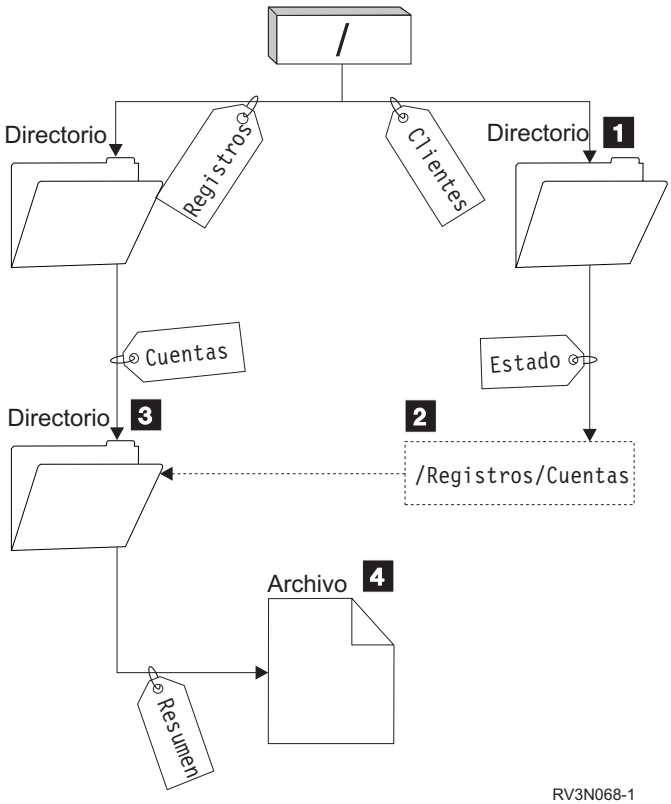

Figura 5. Ejemplo de utilización de un enlace simbólico

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:

```
/Cliente/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 *visión general*, 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, puede crear un enlace simbólico en el sistema de archivos "raíz" (/) o el sistema de archivos QOpenSys que le permita:

- 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.

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.

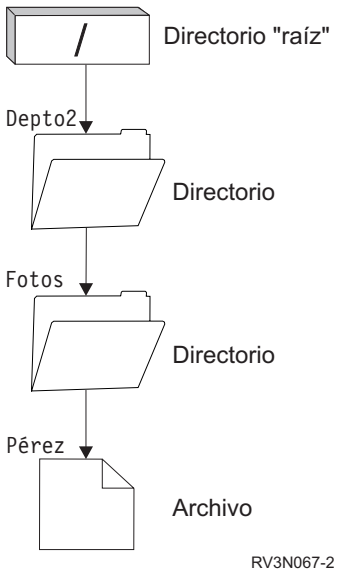

Figura 6. Los componentes de un nombre de vía de acceso

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.

Referencia relacionada

"Nombres de vía de acceso en el sistema de archivos ″raíz″ (/)" en la página 34
Los nombres de vías de acceso tiene un formato específico en el sistema de archivos "raíz" (/).

"Nombres de vía de acceso en el sistema de archivos QOpenSys" en la página 37
Los nombres de vías de acceso tiene un formato específico en el sistema de archivos QOpenSys.400.

"Nombres de vías de acceso en el sistema de archivos QSYS.LIB de ASP independiente" en la página 50
Cada componente del nombre de vía de acceso debe contener el nombre del objeto seguido de su tipo de objeto.

"Nombres de vía de acceso en el sistema de archivos QNTC" en la página 61
La vía de acceso está formada por el nombre de sistema de archivos, el nombre del servidor, el nombre compartido, los nombres del directorio y el subdirectorio, y el nombre del objeto.

"Nombres de vías de acceso en el sistema de archivos QFileSvr.400" en la página 65
Los nombres de vías de acceso tiene un forma específico en el sistema de archivos QFileSvr.400.

"Nombres de vía de acceso en el sistema de archivos QOPT" en la página 55
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.

"Reglas de nombres de vía de acceso para las API" en la página 126
Cuando se utiliza una API del sistema de archivos integrado o de ILE C/400 para realizar operaciones

sobre un objeto, este se identifica especificando su vía de acceso de directorio. A continuación se proporciona una visión general de las reglas que hay que tener en cuenta al especificar nombres de vías de acceso en las API.

"Reglas que rigen los nombres de vía de acceso en los mandatos CL y pantallas" en la página 77
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.

"Nombres de vías de acceso en el sistema de archivos QSYS.LIB" en la página 47
Cada componente del nombre de vía de acceso debe contener el nombre del objeto seguido de su tipo de objeto.

"Nombres de vía de acceso en el sistema de archivos QDLS" en la página 53
Cada componente del nombre de la vía de acceso puede componerse de un solo nombre.

"Nombres de vía de acceso en un sistema de archivos definido por usuario del sistema de archivos integrado" en la página 40
Un objeto archivo especial de bloqueo (*BLKSF) representa un sistema de archivos definido por el usuario (UDFS) cuando hay que manipular la totalidad del UDFS y todos los objetos que hay en él.

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 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 de iSeries. Un archivo de base de datos está orientado a registros; tiene subdivisiones predefinidas que constan de uno o más campos con características específicas, como la longitud y el tipo de datos.

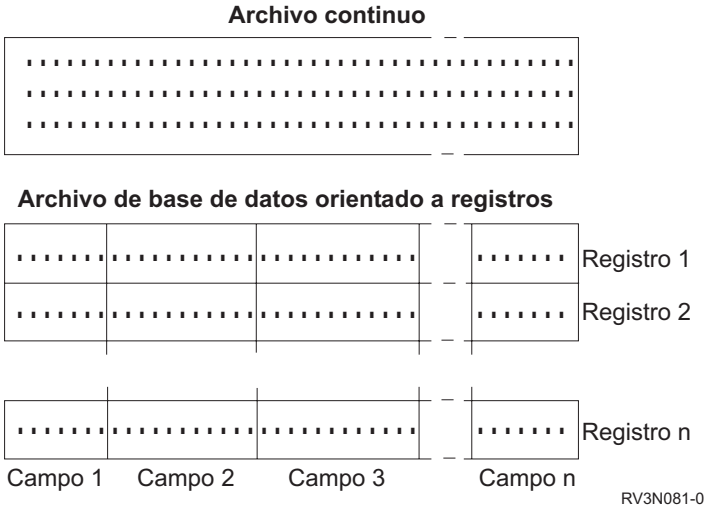

Figura 7. Comparación entre un archivo continuo y un archivo orientado a registros

Los archivos continuos y los archivos orientados a registros están estructurados de forma distinta, y esta diferencia estructural afecta a su utilización. La estructura afecta a cómo se escribe una aplicación para que interaccione con los archivos y también afecta a dónde se utiliza mejor cada tipo de archivo en una aplicación. Un archivo orientado a registros, por ejemplo, resulta adecuado para almacenar datos estadísticos de cliente, como el nombre, la dirección y el saldo de cuentas. Un archivo orientado a registros 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.

Un archivo tiene una de las dos opciones de formato siguientes: archivo continuo *TYPE1 o archivo continuo *TYPE2. El formato del archivo depende del release en el que se creó el archivo, pero si se creó en un sistema de archivos definido por el usuario, depende del valor especificado para dicho sistema de archivos.

Nota: El concepto de archivos continuos *TYPE1 y *TYPE2 es distinto del concepto de formatos de directorios *TYPE1 y *TYPE2. No están relacionados.

Archivos continuos *TYPE1

Un archivo continuo *TYPE1 tiene el mismo formato que los archivos continuos creados en releases anteriores a OS/400 V4R4.

El archivo continuo *TYPE1 tiene un tamaño mínimo de 4096 bytes. Los archivos continuos *TYPE1 tienen un tamaño máximo de objeto de aproximadamente 128 GB (1 GB es igual a aproximadamente 1.073.741.824 bytes).

Archivos continuos *TYPE2

Un archivo continuo *TYPE2 tiene acceso a archivos de alto rendimiento.

Los archivos continuos *TYPE2 tienen un tamaño máximo de objeto de aproximadamente 1 TB (1 TB equivale aproximadamente a 1.099.511.627.776 de bytes) en los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario. De lo contrario, el máximo es aproximadamente 256 GB. También permite la correlación de memoria y la especificación de un atributo para optimizar la asignación del almacenamiento principal. Todos los archivos creados con OS/400 V4R4 y sistemas más nuevos son archivos continuos *TYPE2, a menos que se crearan en un sistema de archivos definido por usuario que especificara un formato de archivo *TYPE1.

Nota: Los archivos mayores de 256 GB no se pueden salvar ni restaurar en sistemas anteriores a i5/OS V5R3.

Referencia relacionada

“Sistemas de archivos definidos por el usuario (UDFS)” en la página 38
Los sistemas de archivos definidos por el usuario (UDFS) residen en las agrupaciones de almacenamiento auxiliar (ASP) o en las agrupaciones de almacenamiento auxiliar independientes (IASP) que se elijan. Estos sistemas de archivos los crea y los gestiona el usuario.

“Copia de datos entre archivos continuos y archivos de base de datos” en la página 113
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.

Continuidad de nombres

Si utiliza sistemas de archivos “raíz” (/), QOpenSys y definidos 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 los servidores iSeries 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. 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.

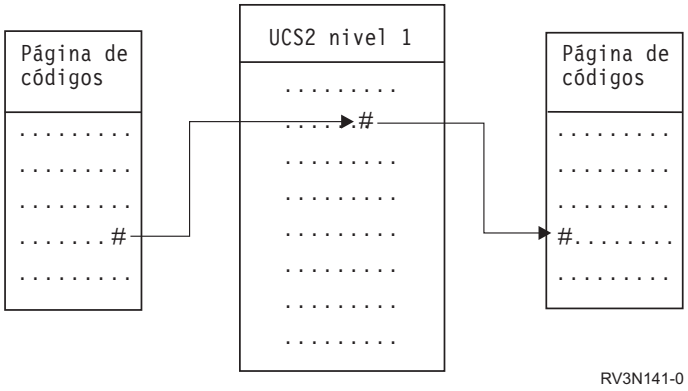

Figura 8. Mantener los caracteres igual en los esquemas de codificación

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.

Conceptos relacionados

"Directorios *TYPE2" en la página 10
Los sistemas de archivos "raíz" (/), QOpenSys y 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.

Referencia relacionada

"Nombres de vía de acceso en el sistema de archivos "raíz" (/)" en la página 34
Los nombres de vías de acceso tiene un formato específico en el sistema de archivos "raíz" (/).

"Nombres de vía de acceso en un sistema de archivos definido por usuario del sistema de archivos integrado" en la página 40
Un objeto archivo especial de bloqueo (*BLKSF) representa un sistema de archivos definido por el usuario (UDFS) cuando hay que manipular la totalidad del UDFS y todos los objetos que hay en él.

"Nombres de vías de acceso en el sistema de archivos QFileSvr.400" en la página 65
Los nombres de vías de acceso tiene un forma específico en el sistema de archivos QFileSvr.400.

"Soporte internacional y asignación de nombres" en la página 129
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.

Atributos ampliados

Un atributo ampliado es la información asociada a un objeto que facilita detalles adicionales acerca del mismo. El atributo ampliado 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 atributos ampliados de un objeto existen solo mientras el objeto existe.

Existen muchos tipos de atributos ampliados, que se pueden utilizar para contener una gran variedad de información. En particular, puede que necesite tener en cuenta los tres atributos ampliados 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 atributo ampliado asociado al objeto.

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

Diversos objetos en los distintos sistemas de archivos pueden tener o no atributos ampliados. 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 atributo ampliado. Algunas carpetas y documentos pueden tener atributos ampliados y puede que otras no. En los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario, todos los directorios, archivos continuos y enlaces simbólicos pueden tener atributos ampliados de cualquier tipo. Algunos, no obstante, puede que no tengan ningún atributo ampliado.

Los mandatos Trabajar con enlaces de objeto (WRKLNK) y Visualizar enlaces de objetos (DSPLNK) se pueden utilizar para visualizar el atributo ampliado .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 atributos ampliados. 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 atributos ampliados asociados a algunos objetos de QDLS pueden cambiarse mediante interfaces proporcionadas por el sistema de archivos jerarquizado (HFS).

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 (por ejemplo DosQueryFileInfo y DosSetFileInfo) para consultar y establecer los atributos ampliados de cualquier objeto de archivo. Los usuarios de OS/2 también pueden modificar los atributos ampliados 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 atributo ampliado puede tener una longitud máxima de 255 caracteres.
- No utilice un punto (.) como primer carácter del nombre. Un atributo ampliado cuyo nombre empieza con un punto se interpreta como un atributo ampliado del sistema estándar.
- Para minimizar la posibilidad de conflictos de nombres, utilice una estructura de denominación coherente para los atributos ampliados. Se aconseja el formato siguiente:

  ```
  NombreEmpresaNombreProducto.Nombre_Atributo
  ```

Soporte para la exploración

Con iSeries, puede explorar objetos del sistema de archivos integrado.

Este soporte ofrece una mayor flexibilidad a los usuarios de iSeries al permitir exploraciones según diversos elementos; los usuarios deciden cuándo debe producirse la exploración y qué acciones hay que realizar según los resultados de las exploraciones.

Los dos puntos de salida relacionados con este soporte son:

- QIBM_QP0L_SCAN_OPEN - Programa de salida de exploración del sistema de archivos integrado al abrir

  Para este punto de salida, se llama al programa de salida de exploración del sistema de archivos integrado al abrir para que lleve a cabo un proceso de exploración cuando se abra un objeto del sistema de archivos integrado bajo ciertas condiciones.
- QIBM_QP0L_SCAN_CLOSE - Programa de salida de exploración del sistema de archivos integrado al cerrar

  Para este punto de salida, se llama al programa de salida de exploración del sistema de archivos integrado al cerrar para que lleve a cabo un proceso de exploración cuando se cierre un objeto del sistema de archivos integrado bajo ciertas condiciones.

Nota: Solo se explorarán los objetos de sistemas de archivos que se hayan convertido completamente a directorios *TYPE2.

Tareas relacionadas

"Establecer si debe o no explorarse los objetos" en la página 141
Siga estos pasos para establecer si un objeto debe o no explorarse.

Referencia relacionada

"Exploración del sistema de archivos integrado" en la página 102
Los objetos de los sistemas de archivos "raíz" (/), QOpenSys y UDFS de la ASP de usuario no se explorarán utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado hasta que los sistemas de archivos se hayan convertido por completo al formato de directorios *TYPE2.

Información relacionada

QIBM_QP0L_SCAN_OPEN

QIBM_QP0L_SCAN_CLOSE

Ejemplos: exploración de virus y los archivos que se están abriendo

Estos ejemplos muestran los elementos que el programa de salida puede explorar.

- Virus

  Los programas de salida pueden explorar en busca de virus. Si se encuentra un virus en un archivos, el programa antivirus puede actuar del modo pertinente, reparando el problema o intentando dejar el virus en cuarentena. Puesto que el propio servidor iSeries no estaría infectado por el virus, lo que se lograría es una reducción en las transmisiones de virus entre servidores.
- Llamadas para saber cuándo se abrió un archivo

  También se puede explorar para averiguar cuándo se abrió un archivo. Mediante esta exploración, se puede averiguar la fecha y la hora en las que se accedió a determinados archivos. Le resultará útil si desea controlar el comportamiento de determinados usuarios.

La exploración puede producirse en dos momentos diferentes, según cómo se hayan establecido los valores del sistema y el entorno de exploración. En la lista siguiente se describen los distintos tipos de exploración, dependiendo de cuándo se produzcan.

1. Exploración en tiempo de ejecución

   Una exploración en tiempo de ejecución es una exploración de un archivo o archivos durante las actividades diarias normales. Esto garantiza la integridad de sus archivos cada vez que accede a ellos. Las exploración durante actividades normales le permite asegurarse que su archivo o archivos son actuales para los estándares que se utilizan para la exploración.

   Ejemplo de exploración en tiempo real en búsqueda de virus

   Supongamos que desea acceder a un archivo del sistema de archivos integrado desde su ordenador. Cuando se abre el archivo desde el ordenador, este archivo se explora. Esto se debe a que se ha registrado un programa de salida de apertura y el valor del sistema QSCANFS se ha definido para explorar los archivos de los sistemas de archivos "raíz" (/), QOpenSys y UDFS. La exploración muestra que se ha encontrado un virus y el programa de salida antivirus procede a reparar el

problema. Cuando el programa de salida repara el archivo, el archivo ya no está infectado. Por lo tanto, el acceso desde el PC no está infectado y no puede propagar la infección.

Supongamos que en lugar de efectuar una exploración para detectar la presencia de virus en dicho acceso, decide no efectuar una exploración en tiempo real. A continuación, después de acceder al archivo infectado desde su ordenador, el virus puede pasar al ordenador. Si efectuara una exploración en tiempo real, el virus se puede detectar antes de que se propague por su PC.

El principal inconveniente de este método es que para las exploraciones se necesita tiempo de recursos. Aquellos usuarios que intenten acceder a un archivo deberán esperar hasta que finalice la exploración, antes de poder utilizar el archivo. El sistema garantiza que se realiza la exploración solo cuando es necesario, no en cada acceso.

2. Exploración en masa o activada manualmente

   Puede utilizar esta opción si desea explorar diversos elementos simultáneamente. En ese caso, puede establecer la exploración para que se produzca cuando el servidor esté inactivo, por ejemplo, durante el fin de semana. De esta forma, afectaría poco al acceso a archivos durante las actividades diarias habituales. La exploración se realiza fuera de línea. Por lo tanto, se pueden reducir los costes operativos de la exploración en tiempo de ejecución de los archivos que no cambian cuando termina la exploración en masa. Esto se debe a que no es necesario repetir las exploraciones cuando se vuelve a acceder a estos archivos.

Conceptos relacionados

"Valores del sistema relacionados"
Existen dos valores del sistema relacionados con este soporte de exploración. Puede utilizar estos dos valores del sistema para definir el entorno de exploración que desea para su servidor.

Información relacionada

QIBM_QP0L_SCAN_OPEN

QIBM_QP0L_SCAN_CLOSE

Valores del sistema relacionados

Existen dos valores del sistema relacionados con este soporte de exploración. Puede utilizar estos dos valores del sistema para definir el entorno de exploración que desea para su servidor.

A continuación encontrará los nombres de los dos valores del sistema y las descripciones para cada uno de ellos. Se describen estos valores del sistema y sus opciones de control para el iSeries Navigator. Los valores de la interfaz basada en caracteres comparables aparecen entre paréntesis después de los nombres del iSeries Navigator. Por ejemplo, para el valor del sistema QSCANFSCTL, cuando se selecciona la opción de control ″Explorar accesos solo mediante servidores de archivos″ de iSeries Navigator, estaría creando básicamente los mismos resultados que si especificase la opción de control basada en caracteres *FSVRONLY.

Los nombres y descripciones de estos valores del sistema son los siguientes:

1. Utilizar programa de salida registrado para explorar los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario (QSCANFS).

   Este valor del sistema puede utilizarse para especificar si los sistemas de archivos deben o no explorarse. Si el sistema de archivos se ha convertido totalmente, solo se explorarán los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario. Este valor especifica si los objetos deberían explorarse mediante programas de salida registrados o mediante cualquiera de los puntos de salida relacionados con la exploración del sistema de archivos integrado.

   El valor por omisión es que los objetos se explorarán si se ha registrado algún programa de salida.

2. Control de exploración (QSCANFSCTL)

   Para este valor del sistema, puede utilizar las opciones de control por omisión o puede utilizar las opciones de control especificadas. Para obtener breves descripciones de las diferentes opciones de control especificadas basadas en los valores del sistema de iSeries Navigator, consulte más abajo.

   - Explorar accesos solo mediante servidores de archivos - (se ha especificado *FSVRONLY)

La exploración solo tendrá lugar si se accede al iSeries desde un servidor de archivos. Si no se selecciona esta opción, se explorarán todos los accesos.

- Finalizar petición si el programa de salida falla - (se ha especificado *ERRFAIL)

  Si existen errores al llamar al programa de salida, fallará la petición o la operación que desencadenó la llamada al programa de salida. Si no se selecciona esta opción, el sistema omitirá el programa de salida anómalo y el objeto se tratará como si no se hubiera explorado.

- Realizar actualizaciones de acceso de escritura - (no se ha especificado *NOWRTUPG)

  La actualización del acceso se producirá para el descriptor de exploración transmitido al programa de salida para incluir el acceso de escritura. Si no se selecciona la opción *NOWRTUPG, el sistema no intentará efectuar la actualización del acceso de escritura.

  si se ha especificado *NOWRTUPG, el sistema **no** intentará actualizar el acceso para el descriptor de exploración transmitido al programa de salida para incluir el acceso de escritura. Si no se ha especificado *NOWRTUPG, el sistema intentará efectuar la actualización del acceso de escritura.

- Utilizar el atributo ″solo cuando haya objetos cambiados″ para controlar la exploración - (se ha especificado *USEOCOATR)

  Se utilizará el atributo ″solo cambio de objeto″ (explorar solo el objeto si ha sido modificado). Si no se ha seleccionado esta opción, este atributo no se utilizará y el objeto se explorará tras ser modificado y cuando el software de exploración indique una actualización.

- Finalizar petición de cierre si la exploración finaliza de forma anómala durante el cierre (no se ha especificado *NOFAILCLO)

  La petición de cierre finalizará de forma anómala si finalizó de forma anómala la exploración de un objeto durante el proceso de cierre. Si no se selecciona esta opción, no se finalizará de forma anómala la petición de cierre. Si no se selecciona, este valor altera temporalmente la especificación del valor ″finalizar petición si el programa de salida falla″.

  Si se especifica *NOFAILCLO, el sistema **no** finalizará la petición de cierre con una indicación de anomalía de exploración, aunque haya fallado una exploración del objeto realizada como parte del proceso de cierre.

- Explorar en siguiente acceso tras la restauración del objeto - (no se ha especificado *NOPOSTRST)

  Los objetos se explorarán tras ser restaurados. Si se especifica el atributo ″el objeto no se explorará″, el objeto se explorará una vez tras su restauración. Si se especifica el atributo ″solo cambio de objeto″, el objeto se explorará tras su restauración.

  Si se especificó *NOPOSTRST durante la restauración de los objetos, estos no se explorarán solo por haber sido restaurados. Si el atributo del objeto es ″el objeto no se explorará″, el objeto no se explorará en ningún momento. Si el atributo del objeto es ″solo cambio de objeto″, el objeto solo se explorará si se modifica tras su restauración.

Referencia relacionada

“Exploración del sistema de archivos integrado” en la página 102
Los objetos de los sistemas de archivos ″raíz″ (/), QOpenSys y UDFS de la ASP de usuario no se explorarán utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado hasta que los sistemas de archivos se hayan convertido por completo al formato de directorios *TYPE2.

Información relacionada

QSCANFS

QSCANFSCTL

Apariciones de la exploración

La exploración puede producirse por diversos motivos. En este apartado se indica cuándo y por qué podría producirse una exploración.

Para ver el estado y el atributo de exploración actuales para un objeto, puede utilizar el mandato Trabajar con enlaces de objetos (WRKLNK), el mandato Visualizar enlaces de objetos (DSPLNK), la API Obtener atributos (Qp0lGetAttr()) o la página Propiedades de iSeries Navigator.

Información relacionada

Mandato Trabajar con enlaces de objetos (WRKLNK)

Mandato Visualizar enlaces de objetos (DSPLNK)

API Obtener atributos (QP01GetAttr())

Modificación de objetos:

Se producirá una exploración si se accede a un objeto tras haber sido modificado.

Normalmente, la modificación se producirá en los datos del objeto. Ejemplos de modificaciones de un objeto son: escribir en el objeto directamente o mediante correlaciones de memoria, truncar el objeto o borrarlo. Si se cambia el atributo CCSID del objeto, también se desencadena una exploración en el próximo acceso.

Cambio de firma:

Se producirá una exploración cuando se acceda al objeto si la firma global es diferente de la firma del objeto.

Las firmas del grupo de ASP globales o independientes representan el nivel de software asociado con los programas de salida relacionados con la exploración. La firma del objeto refleja la firma de la ASP global o independiente la última vez en que se exploró el objeto. Si un objeto no está en un grupo de ASP independientes, la firma del objeto se compara con la firma de exploración global. Si el objeto está en una ASP independiente, la firma del objeto se compara con la firma de exploración del grupo de ASP independientes asociado.

Nota: En el siguiente ejemplo, se utilizan los conceptos clave de exploración y firma de clave de exploración. La clave de exploración es un método para identificar un conjunto de software de exploración. Por ejemplo, para una empresa específica. La firma de la clave de exploración permite al conjunto de software de exploración indicar el nivel de soporte que proporciona. Por ejemplo, un conjunto de definiciones de virus.

A continuación aparece un ejemplo de qué sucede cuándo un objeto no se encuentra en un grupo de ASP independientes y se lleva a cabo una exploración:

1. Se registra un programa de salida en el punto de salida QIBM_QP0L_SCAN_OPEN. Se han especificado una clave de exploración y una firma de clave de exploración, del siguiente modo:

   Clave de exploración: XXXXXX
   Firma de la clave de exploración: 0000000000

   La firma de exploración global es 0000 y no se ha actualizado.
2. Luego se registra un programa de salida en el punto de salida QIBM_QP0L_SCAN_CLOSE. Se han especificado una clave de exploración y una firma de clave de exploración, del siguiente modo:

   Clave de exploración: XXXXXX
   Firma de la clave de exploración: 1111111111

   Posteriormente se actualiza la firma de exploración global a 0001.
3. A continuación, se abre un archivo que actualmente posee una firma de objeto de 0000. La existencia de los programas de salida, junto con la diferencia en las firmas de exploración global (0000 y 0001), inicia una exploración. Si la exploración finaliza satisfactoriamente, la firma del archivo se actualiza a 0001.
4. Si otro usuario abre el archivo, este no volverá a explorarse, pues las firmas global y del objeto concuerdan.

El siguiente ejemplo muestra una situación en la que el programa de salida se propone provocar una nueva exploración.

1. Se ha añadido soporte al sistema para explorar en busca de nuevos tipos de virus. Se llama a la API Cambiar firma de exploración (QP0LCHSG) para actualizar la firma de la clave de exploración. Se especifica una clave de exploración y una firma de clave de exploración, del siguiente modo:

   Clave de exploración: XXXXXX
   Firma de la clave de exploración: 2222222222

   Posteriormente se actualiza la firma de la clave de exploración global a 0002.
2. Si se abriera ahora el archivo explorado anteriormente, la diferencia en las firmas provocaría una nueva exploración.

El ejemplo continúa para mostrar qué sucede si el objeto se encuentra en grupo de ASP independientes:

1. Se activa una ASP independiente por primera vez y se abre un archivo de la ASP independiente. Cuando se abre el primer archivo, se compara la lista de claves de exploración de la ASP independiente con la lista de claves de exploración del sistema. Las dos son distintas, puesto que no existe ninguna lista de claves de exploración para la ASP independiente. En ese caso, la lista de claves de exploración de la ASP independiente obtiene la lista de claves de exploración global. La lista de claves de exploración de la ASP independiente tendría entonces una clave de exploración XXXXXX y una firma de clave de exploración 2222222222. Como resultado, se cambia la firma de exploración de la ASP independiente a 0001. Cuando se abre el archivo en la ASP independiente que actualmente posee una firma de objeto de 0000, este se compara con la firma de exploración de la ASP independiente 0001 y esta diferencia desencadena la exploración del archivo. Si la exploración finaliza satisfactoriamente, la firma del archivo se actualiza a 0001.

Nota: Un cambio de firma desencadenaría una exploración a menos que para el objeto se hayan especificado el atributo ″solo cambio de objeto″ y el valor del sistema *USEOCOATR.

Información relacionada

QIBM_QP0L_SCAN_OPEN

QIBM_QP0L_SCAN_CLOSE

API Cambiar firma de exploración (QP0LCHSG)

CCSID diferente:

Si se accede a un objeto con un identificador de juego de caracteres (CCSID) distinto del utilizado en la exploración anterior de ese objeto, se desencadenará una exploración.

Un ejemplo de esta exploración es cuando un archivo con datos almacenado con el CCSID 819 se abre con el CCSID 1200 y la exploración es satisfactoria. Mientras no se modifiquen los datos del archivo, no se producirá una exploración cada vez que el archivo se abra en el CCSID 1200. Sin embargo, si dicho archivo se abriera en un CCSID distinto, por ejemplo 37, se desencadenaría una exploración para dicho CCSID 37. Si esa exploración también fuera satisfactoria, cualquier acceso posterior con CCSID 1200 y 37 no desencadenarían exploraciones adicionales.

Solo se mantienen dos CCSID y una indicación binaria con el objetivo de minimizar los datos almacenados en el sistema. Si normalmente accede al mismo objeto con diversos CCSID, podría desencadenar diversas exploraciones adicionales.

Durante la operación de salvar:

Se trata de otro ejemplo de cuándo podría producirse una exploración. Puede solicitarse una exploración cuando se salva un objeto.

El mandato Salvar objetos (SAV) incluye ahora un parámetro SCAN que permite especificar si deben explorarse los archivos en el momento en que se salven. También puede solicitar que no se salve el objeto si antes no ha superado con éxito un proceso de exploración o si no supera con éxito la exploración realizada en el momento de salvarlo. Esto evita que se salven en medios los archivos que no superan la exploración y que puedan traspasarse a otros sistemas.

Nota: Ello no significa que cuando se restaure dicho objeto se marque como explorado. Cuando se restauran objetos, se borra todo el historial del estado de exploración.

Información relacionada

Mandato Salvar objeto (SAV)

Comprobar integridad del objeto:

En último lugar, puede solicitarse una exploración si se especifica el parámetro SCANFS en el mandato Comprobar integridad del objeto(CHKOBJITG) con un valor de *YES.

Esta opción es ideal cuando se desea determinar la validez de un archivo sin abrirlo. Si se especifica SCANFS (*STATUS), todos los objetos que no han superado con éxito exploraciones anteriores registran una violación de fallo de exploración.

Información relacionada

Mandato Cambiar integridad del objeto (CHGOBJITG)

Trabajar con sistemas de archivos

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 interaccionar 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 de i5/OS 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 de i5/OS 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 interaccionar con cualquiera de los sistemas de archivos mediante una interfaz común. Esta interfaz está optimizada para entrada y salida de datos continuos, en contraste con la entrada y 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 mediante esta interfaz común.

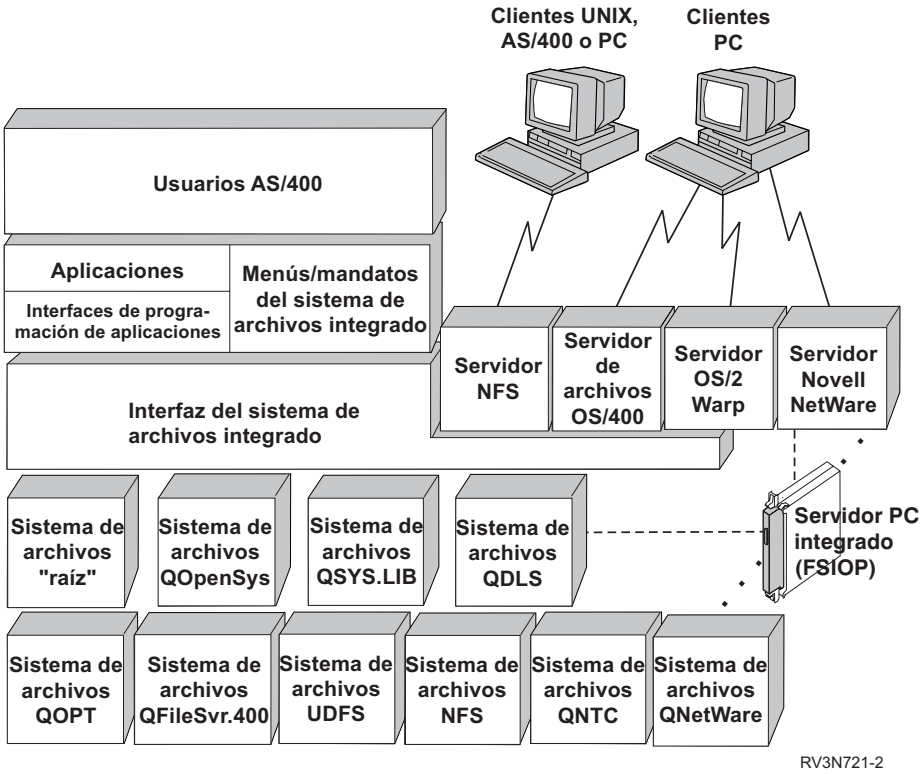

Figura 9. Sistemas de archivos, servidores de archivos y la interfaz del sistema de archivos integrado

Utilización de sistemas de archivos de red en la interfaz del sistema de archivos integrado

Puede acceder al sistema de archivos de red (NFS) a través de la interfaz del sistema de archivos integrado. Tenga en cuenta estas consideraciones y limitaciones.

Conceptos relacionados

"¿Qué es el sistema de archivos integrado?" en la página 2
El *sistema de archivos integrado* es un componente de i5/OS 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.

Tareas relacionadas

"Acceso a los menús y las pantallas" en la página 72
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.

"Trasladar archivos o carpetas a otro sistema de archivos" en la página 136
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.

Referencia relacionada

"Acceso mediante mandatos CL" en la página 73
Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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.

"Reglas que rigen los nombres de vía de acceso en los mandatos CL y pantallas" en la página 77
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.

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Información relacionada

Soporte óptico

Soporte del sistema de archivos de red

Comparación de los sistemas de archivos

En estas tablas se resumen las características y las limitaciones de cada sistema de archivos.

Tabla 2. Visión general de los sistemas de archivos (Parte 1 de 2)

Posibilidad	**"raíz" (/)**	**QOpenSys**	**QSYS.LIB[16]**	**QDLS**	**QNTC**
Componente estándar del i5/OS	Sí	Sí	Sí	Sí	Sí
Tipo de archivo	Continuo	Continuo	Registro [12]	Continuo	Continuo
Límite tamaño archivo	T2=1 TB; T1=128 GB	T2=1 TB; T1=128 GB	Tamaños archivo base de datos	4 GB	Variable [17]
Integrado con OfficeVision (por ejemplo, el archivo se puede enviar por correo)	No	No	No	Sí	No
Acceso a través del servidor de archivos i5/OS	Sí	Sí	Sí	Sí	Sí
Acceso directo a través del procesador de E/S de servidor de archivos [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. [19]	255 cars. [19]	10.6 cars. [6]	8.3 cars. [7]	255 cars. [19]
Longitud máxima de nombre de vía de acceso [8]	16 MB	16 MB	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	0 [18]
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
Admite enlaces simbólicos	Sí	Sí	No	No	No
Objeto/archivo puede tener propietario	Sí	Sí	Sí	Sí	No
Admite mandatos del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Admite las API del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Admite 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 2. Visión general de los sistemas de archivos (Parte 1 de 2) (continuación)

Posibilidad	"raíz" (/)	QOpenSys	QSYS.LIB[16]	QDLS	QNTC

Notas:

1. El procesador de E/S de servidor de archivos es un hardware utilizado por el Servidor LAN.
2. Cuando se accede a través del servidor de archivos i5/OS.
3. Cuando se accede a través de un PC cliente de Servidor LAN. El acceso utilizando las API del iSeries es comparativamente lento.
4. El sistema de archivos QSYS.LIB tiene una longitud máxima de nombre de vía de acceso de 55 caracteres. El sistema de archivos QSYS.LIB de la ASP independiente tiene una longitud máxima de vía de acceso de 66 caracteres.
5. Consulte el apartado "Sistema de archivos de servicios de biblioteca de documentos (QDLS)" en la página 52 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.
7. Un máximo de 8 caracteres en el nombre y de 1 a 3 caracteres en la extensión de tipo de archivo (si procede).
8. Suponiendo un nombre de vía de acceso absoluto que empiece por / seguido del nombre del sistema de archivos (por ejemplo, `/QDLS...`).
9. Los sistemas de archivos QSYS.LIB y QSYS.LIB de ASP independiente 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 solo puede tener un enlace con otro directorio.
12. Los espacios de usuario de QSYS.LIB y QSYS.LIB de ASP independiente 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. Los sistemas de archivos QSYS.LIB y QSYS.LIB de ASP independiente pueden dar soporte al registro por diario de objetos distintos de los sistemas de archivos "raíz" (/), UDFS y QOpenSys.
15. Los directorios *TYPE2 tienen un límite de un millón de enlaces por objeto y un límite de 999.998 subdirectorios. Los directorios *TYPE1 tienen un límite de 32.767 enlaces por objeto.
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.
17. Depende del sistema al que se accede.
18. QNTC no soporta los atributos ampliados.
19. Para algunos valores de CCSID, la longitud máxima puede ser inferior a 255 caracteres.

Abreviaturas

- cars. = caracteres
- T1 = *TYPE1 *STMF
- T2 = *TYPE2 *STMF
- B = bytes KB = kilobytes MB = megabytes GB = gigabytes TB = terabytes

Tabla 3. Visión general de los sistemas de archivos (Parte 2 de 2)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Componente estándar del i5/OS	Sí	Sí	Sí	Sí	No
Tipo de archivo	Continuo	Continuo	Continuo	Continuo	Continuo
Límite tamaño archivo	4 GB	2 GB - 1	T2 = 1 TB; T1=128 GB	Variable [16]	2 GB

Tabla 3. Visión general de los sistemas de archivos (Parte 2 de 2) (continuación)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Integrado con OfficeVision (por ejemplo, el archivo se puede enviar por correo)	No	No	No	No	No
Acceso a través del servidor de archivos i5/OS	Sí	Sí	Sí	Sí	Sí
Acceso directo a través del Servidor PC Integrado [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. [17]	Variable [2]	255 cars. [13, 17]
Longitud máxima de nombre de vía de acceso	294 cars.	Sin límite [2]	16 MB	Sin límite [2]	255 cars.
Longitud máxima de los atributos ampliados de un objeto	8 MB	0 [6]	2 GB [10]	0 [6]	64 KB
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
Admite enlaces simbólicos	No	No	Sí	Sí [2]	No
Objeto/archivo puede tener propietario	No	No [9]	Sí	Sí [2]	Sí
Admite mandatos del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Admite las API del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Admite las API del sistema de archivos jerarquizado (HFS)	Sí	No	No	No [2]	No
Admite ejecución en hebras [14]	Sí	Sí	Sí	Sí	No
Admite el registro por diario de objetos	No	No	Sí	No	No

Tabla 3. Visión general de los sistemas de archivos (Parte 2 de 2) (continuación)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare

Notas:

1. El procesador de E/S de servidor de archivos es un hardware utilizado por el Servidor LAN.
2. Depende de a qué sistema de archivos remoto se esté accediendo.
3. Cuando se accede a través del servidor de archivos i5/OS.
4. Consulte el apartado "Sistema de archivos óptico (QOPT)" en la página 54 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 solo 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.
16. Depende del sistema al que se accede.
17. Para algunos valores de CCSID, la longitud máxima puede ser inferior a 255 caracteres.

Abreviaturas

- cars. = caracteres
- T1 = *TYPE1 *STMF
- T2 = *TYPE2 *STMF
- B = bytes KB = kilobytes MB = megabytes GB = gigabytes TB = terabytes

Tareas relacionadas

"Trasladar archivos o carpetas a otro sistema de archivos" en la página 136
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.

Referencia relacionada

"Sistema de archivos ″raíz″ (/)" en la página 33
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. Tiene las características de los sistemas de archivos DOS y OS/2.

"Sistema de archivos de sistemas abiertos (QOpenSys)" en la página 36
El sistema de archivos QOpenSys es compatible con los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX y X/Open Portability Guide (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.

"Sistemas de archivos definidos por el usuario (UDFS)" en la página 38
Los sistemas de archivos definidos por el usuario (UDFS) residen en las agrupaciones de almacenamiento auxiliar (ASP) o en las agrupaciones de almacenamiento auxiliar independientes (IASP) que se elijan. Estos sistemas de archivos los crea y los gestiona el usuario.

"Sistema de archivos de biblioteca (QSYS.LIB)" en la página 45
El sistema de archivos QSYS.LIB da soporte a la estructura de bibliotecas del servidor iSeries.

"QSYS.LIB de ASP independiente" en la página 48
El sistema de archivos QSYS.LIB de ASP independiente ofrece soporte para la estructura de bibliotecas del servidor iSeries 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 servidor iSeries que gestiona el soporte de bibliotecas en las ASP independientes.

"Sistema de archivos de servicios de biblioteca de documentos (QDLS)" en la página 52
El sistema de archivos QDLS soporta la estructura de carpetas. Proporciona acceso a documentos y carpetas.

"Sistema de archivos óptico (QOPT)" en la página 54
El sistema de archivos QOPT da acceso a los datos continuos almacenados en soportes ópticos.

"Sistema de archivos NetWare (QNetWare)" en la página 57
El sistema de archivos QNetWare proporciona acceso a los datos en servidores de PC autónomos que ejecutan Novell NetWare 5.1 o 6.0.

"Sistema de archivos iSeries NetClient (QNTC)" en la página 60
El sistema de archivos QNTC proporciona acceso a los datos y objetos que se encuentran almacenados en un Integrated xSeries Server para iSeries que ejecuta Windows NT 4.0 Server o superior, o Linux. El sistema de archivos QNTC proporciona acceso a los datos y objetos que se encuentran almacenados en servidores remotos que ejecutan Windows NT 4.0 o posterior, Linux Samba 3.0 o posterior, o versiones soportadas de iSeries NetServer.

"Sistema de archivos del servidor de archivos i5/OS (QFileSvr.400)" en la página 65
El sistema de archivos QFileSvr.400 proporciona acceso transparente a otros sistemas de archivos que residen en servidores iSeries remotos. Se accede a él mediante una estructura de directorios jerárquica.

"Sistema de archivos de red (NFS)" en la página 69
El sistema de archivos de red (NFS) proporciona al usuario acceso a los datos y objetos almacenados en un servidor NFS remoto.

Información relacionada

Gestión por diario

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. Tiene las características de los sistemas de archivos DOS y OS/2.

Además:

- Está optimizado para la entrada y salida de archivos continuos.
- Soporta múltiples enlaces fijos y enlaces simbólicos.
- Soporta sockets locales.
- Soporta las API que permiten 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 por diario de los cambios de los objetos.
- Soporta la exploración de objetos utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado.

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 escriban en /dev/null o /dev/zero se descartarán. 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 borrados y sustituidos por ceros.

Utilización del sistema de archivos ″raíz″ (/)

Es posible acceder a un sistema de archivos "raíz" (/) a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

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 vías de acceso tiene un formato específico en el sistema de archivos "raíz" (/).

```
/Directory/Directory . . . /Object
```

- Cada componente del nombre de vía de acceso puede tener una longitud máxima de 255 caracteres, muy superior de la de los sistemas de archivos QSYS.LIB o QDLS. El nombre completo de la vía de acceso puede ser muy extenso, hasta 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 almacenan los nombres.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

"Directorios *TYPE2" en la página 10
Los sistemas de archivos ″raíz″ (/), QOpenSys y 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.

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Todos los mandatos listados en el tema Acceso mediante mandatos CL y las pantallas que se describen en el tema Acceso a los menús y las pantallas pueden realizar operaciones en el sistema de archivos "raíz" (/). Sin embargo, puede que la ejecución de estos mandatos en un proceso que admite la ejecución multihebra no sea segura.

Tareas relacionadas

"Acceso a los menús y las pantallas" en la página 72
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.

Referencia relacionada

"Acceso mediante mandatos CL" en la página 73
Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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.

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

Todas las API que se listan en el tema Realizar operaciones utilizando API pueden operar en el sistema de archivos "raíz" (/).

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Información relacionada

Interfaces de programación de aplicaciones (API)

Registrar por diario los cambios de los objetos en el sistema de archivos ″raíz″ (/)

Los objetos del sistema de archivos "raíz" (/) pueden registrarse por diario. Esta función permite recuperar los cambios realizados en un objeto desde que se guardó por última vez.

Conceptos relacionados

"Registro por diario de objetos" en la página 102
La finalidad primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se salvó por última vez. Además, un uso importante del registro por diario es duplicar en otro sistema los cambios realizados en un objeto, para mejorar la disponibilidad o equilibrar las cargas de trabajo.

Dispositivos UDP y TCP en el sistema de archivos ″raíz″ (/)

El sistema de archivos "raíz" (/) bajo el directorio /dev/xti mantendrá ahora dos controladores de dispositivos denominados udp y tcp.

Ambos controladores son archivos especiales de caracteres (*CHRSFs) y se crean durante la primera carga inicial de programa (IPL). Los controladores de dispositivos UDP y TCP se utilizan para abrir una conexión con los proveedores de transporte UDP y TCP. Ambos controladores son dispositivos de usuario y reciben un número principal de dispositivo nuevo. También tienen aperturas clonadas, lo que significa que cada apertura obtiene una instancia única del dispositivo. El uso de estos dispositivos solo está soportado en PASE (Portable Application Solutions Environment) de i5/OS. La tabla siguiente contiene los objetos que se crearán y sus propiedades.

Tabla 4. Objetos y propiedades del controlador de dispositivos

Nombre de vía de acceso	Tipo	Principal	Secundario	Propietario	Autorizaciones de datos de propietario	Grupo	Autorizaciones de datos de grupo	Autorizaciones de datos públicos
/dev/xti	*DIR	N/P	N/P	QSYS	*RWX	Ninguno	*RX	*RX
/dev/xti/tcp	*CHRSF	Clon	TCP	QSYS	*RW	Ninguno	*RW	*RW
/dev/xti/udp	*CHRSF	Clon	UDP	QSYS	*RW	Ninguno	*RW	*RW

Conceptos relacionados

"Directorios proporcionados" en la página 7
El sistema de archivos integrado crea estos directorios cuando se reinicia el sistema, si todavía no existen.

Información relacionada

PASE (Portable Applications Solutions Environment)

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 X/Open Portability Guide (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 similar a la de los sistemas UNIX.
- Está optimizado para la entrada y salida de archivos continuos.
- Soporta múltiples enlaces fijos y enlaces simbólicos.
- Soporta nombres sensibles a las mayúsculas y minúsculas.
- Soporta sockets locales.
- Soporta las API que permiten ejecución multihebra.
- Soporta objetos *FIFO.
- Ofrece soporte para el registro por diario de los cambios de los objetos.
- Soporta la exploración de objetos utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado.

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.

Utilización de QOpenSys

Es posible acceder a QOpenSys a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

Conceptos relacionados

"Directorios proporcionados" en la página 7
El sistema de archivos integrado crea estos directorios cuando se reinicia el sistema, si todavía no existen.

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 `Sueldo`, otro objeto denominado `SUeldo` y otro denominado `SUELDO` en el mismo directorio de QOpenSys.

Nombres de vía de acceso en el sistema de archivos QOpenSys

Los nombres de vías de acceso tiene un formato específico en el sistema de archivos QOpenSys.400.

```
/QOpenSys/Directory/Directory/ . . . /Object
```

- 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 MB.
- 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 almacenan los nombres.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

"Directorios *TYPE2" en la página 10
Los sistemas de archivos "raíz" (/), QOpenSys y 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.

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Todos los mandatos listados en Acceso mediante mandatos CL y las pantallas que se describen en Acceso a los menús y las pantallas 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.

Tareas relacionadas

"Acceso a los menús y las pantallas" en la página 72
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.

Referencia relacionada

"Acceso mediante mandatos CL" en la página 73
Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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.

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

Todas las API que se listan en el tema Realizar operaciones utilizando API pueden operar en el sistema de archivos QOpenSys.

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Información relacionada

Interfaces de programación de aplicaciones (API)

Registrar por diario los cambios de los objetos en un sistema de archivos QOpenSys

Los objetos del sistema de archivos QOpenSys pueden registrarse por diario. Esta función permite recuperar los cambios realizados en un objeto desde que se salvó por última vez.

Conceptos relacionados

"Registro por diario de objetos" en la página 102
La finalidad primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se salvó por última vez. Además, un uso importante del registro por diario es duplicar en otro sistema los cambios realizados en un objeto, para mejorar la disponibilidad o equilibrar las cargas de trabajo.

Sistemas de archivos definidos por el usuario (UDFS)

Los sistemas de archivos definidos por el usuario (UDFS) residen en las agrupaciones de almacenamiento auxiliar (ASP) o en las agrupaciones de almacenamiento auxiliar independientes (IASP) que se elijan. 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 optimizado para la entrada y salida de archivos continuos
- Soportan múltiples enlaces fijos y enlaces simbólicos
- Soportan sockets locales
- Soporta las API que permiten ejecución multihebra
- Soporta objetos *FIFO
- Soporta el registro por diario de los cambios de los objetos
- Soporta la exploración de objetos utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado

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.
- El atributo crear exploración de objeto que define cuál debe ser el atributo de exploración para los objetos creados en un UDFS.
- El atributo restringir, redenominar y desenlazar.
- El valor de auditoría para un UDFS.
- Los diferentes formatos de archivo continuo: *TYPE1 y *TYPE2.

Conceptos del sistema de archivos definido por el usuario

En un sistema de archivos definido por el usuario (UDFS), al igual que ocurre en los sistemas de archivos "raíz" (/) y QOpenSys, se pueden crear directorios, archivos continuos, enlaces simbólicos, sockets locales y objetos *FIFO.

Un objeto archivo especial de bloqueo (*BLKSF) representa un UDFS. A medida que se van creando sistemas UDFS, también se van creando automáticamente archivos especiales de bloqueo. Solo se puede acceder 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, se puede acceder a los objetos que hay en él. Cuando se desmonta un UDFS, los objetos que hay en él dejan de ser accesibles.

Para poder acceder a los objetos de un UDFS, hay que montarlo 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.Una carca del programa inicial (IPL) del sistema desmonta todos los UDFS. Por lo tanto, los UDFS se deben volver a montar después de toda IPL.

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

Para obtener más información sobre el montaje de sistemas de archivos, consulte OS/400 Network File System Support .

Utilización del sistema de archivos definido por el usuario mediante la interfaz del sistema de archivos integrado

Es posible acceder a un sistema de archivos definido por el usuario (UDFS) mediante la interfaz del sistema de archivos integrado, utilizando el servidor de archivos i5/OS 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.

Conceptos relacionados

"Enlace" en la página 12
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.

"Archivo continuo" en la página 18
Un *archivo continuo* es una secuencia de bytes accesible aleatoriamente, sin ninguna otra estructura impuesta por el sistema.

Información relacionada

Mandato Crear UDFS (CRTUDFS)

Mayúsculas y minúsculas en un sistema de archivos definido por usuario del sistema de archivos integrado

Se puede especificar si los nombres de objeto del sistema de archivos definido por usuario (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 Crear sistema de archivos definido por usuario (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 solo atañe a cómo busca el usuario los nombres en el servidor.

Información relacionada

Mandato Crear sistema de archivos definido por usuario (CRTUDFS)

Nombres de vía de acceso en un sistema de archivos definido por usuario del sistema de archivos integrado

Un objeto archivo especial de bloqueo (*BLKSF) representa un sistema de archivos definido por el usuario (UDFS) cuando hay que manipular la totalidad del UDFS y todos los objetos que hay en él.

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
```

Aquí, `XX` es el número de la ASP en la que 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 MB.
- 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 almacenan los nombres.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

"Directorios *TYPE2" en la página 10
Los sistemas de archivos "raíz" (/), QOpenSys y 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.

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Enlaces en un sistema de archivos definido por usuario del sistema de archivos integrado

Un sistema de archivos definido por el usuario (UDFS) permite múltiples enlaces fijos con el mismo objeto y soporta plenamente los enlaces simbólicos.

Un enlace simbólico puede crear un enlace de un UDFS a un objeto de otro sistema de archivos.

Conceptos relacionados

"Enlace" en la página 12
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.

Utilización de los mandatos del sistema de archivos integrado en un sistema de archivos definido por el usuario

Todos los mandatos listados en el tema Acceso mediante mandatos CL y las pantallas descritas en el apartado Acceso a los menús y las pantallas pueden realizar operaciones en un sistema de archivos definido por el usuario.

Hay algunos mandatos CL que son específicos del UDFS y otros sistemas de archivos montados en general. Están descritos en la tabla siguiente.

Tabla 5. Mandatos CL del sistema de archivos definido por le usuario

Mandato	Descripción
ADDMFS	Añadir sistema de archivos montado. Pone 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. Muestra información acerca de un sistema de archivos definido por el usuario.

Tabla 5. Mandatos CL del sistema de archivos definido por le usuario (continuación)

Mandato	Descripción
MOUNT	Montar un sistema de archivos. Pone 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 de los espacios de nombre de cliente locales.
UNMOUNT	Desmontar un sistema de archivos. Elimina los sistemas de archivos de servidor remoto exportados de los espacios de nombre de cliente locales. Es un alias del mandato RMVMFS.

Nota: para que los mandatos del sistema de archivos integrado puedan realizar operaciones en los objetos almacenados en un UDFS, primero hay que montarlo.

Tareas relacionadas

"Acceso a los menús y las pantallas" en la página 72
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.

Referencia relacionada

"Acceso mediante mandatos CL" en la página 73
Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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.

Utilización de las API del sistema de archivos integrado en un sistema de archivos definido por el usuario

Todas las API que se listan en el tema Realizar operaciones utilizando API pueden realizar operaciones en un sistema de archivos definido por el usuario.

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

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Información relacionada

Interfaces de programación de aplicaciones (API)

Interfaz gráfica de usuario para un sistema de archivos definido por el usuario

iSeries Navigator, una interfaz gráfica de usuario en el PC, proporciona un acceso sencillo y cómodo a sistemas de archivos definidos por el usuario (UDFS).

Esta interfaz permite crear, suprimir, visualizar, montar y desmontar un UDFS desde un cliente Windows.

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

- "Crear un nuevo sistema de archivos definido por el usuario" en la página 140
- "Montar un sistema de archivos definido por el usuario" en la página 140
- "Desmontar un sistema de archivos definido por el usuario" en la página 141

Crear un sistema de archivos definido por usuario del sistema de archivos integrado

El mandato Crear sistema de archivos definido por usuario (CRTUDFS) crea un sistema de archivos que se puede ver mediante los mandatos CL, las API y el espacio de nombres del sistema de archivos integrado.

Los mandatos ADDMFS o MOUNT colocan el sistema de archivos definido por el usuario (UDFS) encima del directorio local ya existente. Puede crearse un UDFS en la ASP o ASP independiente que se prefiera.

También pueden especificarse los siguientes elementos para un UDFS:

- Distinción entre mayúsculas y minúsculas
- Si los objetos creados en el UDFS deben o no explorarse
- El valor de exploración para los objetos creados en el UDFS
- El valor del atributo restrictred, rename and unlink

Información relacionada

Mandato Crear sistema de archivos definido por usuario (CRTUDFS)

Mandato Añadir sistema de archivos montado (ADDMFS)

Suprimir un sistema de archivos definido por usuario del sistema de archivos integrado

El mandato Suprimir sistema de archivos definido por usuario (DLTUDFS) suprime un sistema de archivos definido por usuario (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.

Información relacionada

Mandato Suprimir sistema de archivos definido por usuario (DLTUDFS)

Visualizar un sistema de archivos definido por usuario del sistema de archivos integrado

El mandato Visualizar sistema de archivos definido por usuario (DSPUDFS) presenta los atributos de un sistema de archivos definido por usuario (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.

Información relacionada

Mandato Visualizar sistema de archivos definido por usuario (DSPUDFS)

Mandato Visualizar información de sistema de archivos montado (DSPMFSINF)

Montar un sistema de archivos definido por usuario del sistema de archivos integrado

Los mandatos Añadir sistema de archivos montado (ADDMFS) y MOUNT hacen que los objetos de un sistema de archivos sean accesibles en el espacio de nombres del sistema de archivos integrado.

Para montar un sistema definido por el usuario (UDFS), es necesario especificar `*UDFS` para el parámetro TYPE del mandato ADDMFS.

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

Información relacionada

Mandato Añadir sistema de archivos montado (ADDMFS)

Desmontar un sistema de archivos definido por usuario del sistema de archivos integrado

El mandato UNMOUNT hace que el contenido de un sistema de archivos definido por el usuario (UDFS) resulte inaccesible para las interfaces del sistema de archivos integrado.

Una vez desmontado un UDFS, no se podrá acceder a sus objetos de forma individual. Los mandatos Eliminar sistema de archivos montado (RMVMFS) o UNMOUNT hacen que un sistema de archivos montado sea inaccesible para el 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, este 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.

Información relacionada

Mandato Eliminar sistema de archivos montado (RMVMFS)

Salvar y restaurar un sistema de archivos definido por usuario del sistema de archivos integrado

Puede salvar y restaurar todos los objetos del sistema de archivos definido por el usuario (UDFS), así como las autorizaciones asociadas a ellos.

El mandato Salvar objeto (SAV) permite salvar objetos en un UDFS, mientras que el mandato Restaurar objeto (RST) permite restaurar objetos del UDFS. Ambos mandatos funcionan tanto si el UDFS está montado como si no lo está. Sin embargo, para salvar correctamente los atributos del UDFS, y no solo los objetos dentro del UDFS, es preciso desmontar el UDFS.

Información relacionada

Mandato Salvar objeto (SAV)

Mandato Restaurar objeto (RST)

Registrar por diario los cambios de los objetos en un sistema de archivos definido por el usuario

Los objetos de los sistemas de archivos definidos por el usuario (UDFS) se pueden registrar por diario. Esta función permite recuperar los cambios realizados en un objeto desde que se guardó por última vez.

Conceptos relacionados

"Registro por diario de objetos" en la página 102
La finalidad primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se salvó por última vez. Además, un uso importante del registro por diario es duplicar en otro sistema los cambios realizados en un objeto, para mejorar la disponibilidad o equilibrar las cargas de trabajo.

Sistema de archivos definido por el usuario y agrupaciones de almacenamiento auxiliar (ASP) independientes

Cuando se activa una ASP independiente, se producen diversos cambios en el sistema de archivos "raíz" (/).

Los cambios son:

- Se crea un directorio dentro del directorio /dev para la ASP independiente. El nombre de este directorio coincide con el nombre de la descripción de dispositivo asociada con la ASP. Si este directorio existe antes de la petición de activación y no está vacío, se llevará a cabo la activación pero

no será posible trabajar con ningún UDFS en la ASP. En ese caso, desactive la ASP independiente, cambie el nombre del directorio o borre su contenido y repita la petición de activación.

- En el interior del directorio /dev/asp_name encontrará los objetos archivo especial de bloqueo asociados con todos los UDFS que residen en la ASP independiente. Siempre habrá un UDFS por omisión proporcionado por el sistema. La vía de acceso al archivo especial de bloqueo del UDFS por omisión es: /dev/asp_name/QDEFAULT.UDFS
- El UDFS por omisión se monta encima del directorio /asp_name. No es necesario que el directorio /asp_name exista antes de la petición de activación. Sin embargo, si existe, debe estar vacío. Si no está vacío, la ASP seguirá activa pero no se montará el UDFS por omisión. En ese caso, cambie el nombre del directorio o borre su contenido y repita la petición de activación o utilice el mandato MOUNT para montar el UDFS por omisión.
- Si la ASP independiente es una ASP primaria o secundaria y el UDFS por omisión se montó satisfactoriamente, en ese caso se montará un sistema de archivos adicional. El sistema de archivos QSYS.LIB de la ASP independiente se montará encima de /asp_name/QSYS.LIB.

Nota: Este sistema de archivos no puede montarse o desmontarse independientemente del UDFS por omisión. Siempre se montará o desmontará automáticamente.

Referencia relacionada

"QSYS.LIB de ASP independiente" en la página 48
El sistema de archivos QSYS.LIB de ASP independiente ofrece soporte para la estructura de bibliotecas del servidor iSeries 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 servidor iSeries que gestiona el soporte de bibliotecas en las ASP independientes.

Sistema de archivos de biblioteca (QSYS.LIB)

El sistema de archivos QSYS.LIB da soporte a la estructura de bibliotecas del servidor iSeries.

Este sistema de archivos permite acceder a los archivos de base de datos y al resto de tipos de objeto del servidor iSeries 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 servidor iSeries 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 de i5/OS, 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 interfaz del sistema de archivos integrado.

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

Es posible acceder a un sistema de archivos QSYS.LIB a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

En lo referente al uso de las interfaces del sistema de archivos integrado, deberá tener en cuenta las consideraciones y limitaciones siguientes.

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

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 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 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

A continuación, se proporcionan algunas restricciones que se deben tener en cuenta cuando se manejan 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 tema fcntl().
- Si algún trabajo tiene abierto un miembro de archivo de base de datos, solo un trabajo tiene acceso de escritura a ese miembro de archivo en todo momento. Las demás peticiones solo tienen acceso de lectura.

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

QSYS.LIB soporta operaciones de entrada y salida continuas en objetos espacio de usuario.

Por ejemplo, un programa puede escribir datos continuos en un espacio de usuario y leer datos en un espacio de usuario. El tamaño máximo de un espacio de usuario es 16.776.704 de 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 para salvar archivos en el sistema de archivos QSYS.LIB

El sistema de archivos QSYS.LIB admite las operaciones de E/S continuas para salvar 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 escritura, no se permiten otras instancias abiertas del archivo. Un archivo de salvar **permite** que existan varias instancias abiertas para lectura, a condición de que ningún trabajo tenga más de una instancia abierta del archivo para lectura. Un archivo de salvar no puede abrirse para

acceso de lectura/escritura. Las operaciones de E/S continuas para salvar 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 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 vías 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 inequívocamente el 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, solo 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 de 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 el tema dedicado a la Globalización de i5/OS.

Conceptos relacionados

“Nombre de vía de acceso” en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Enlaces en el sistema de archivos QSYS.LIB

No pueden crearse o almacenarse enlaces simbólicos 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. Por lo tanto, es posible tener un enlace desde un sistema de archivos con soporte para enlaces simbólicos con un objeto del sistema de archivos QSYS.LIB.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos mandatos y pantallas del sistema de archivos integrado son válidos en el sistema de archivos QSYS.LIB.

Los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QSYS.LIB, con las siguientes restricciones:

- El mandato ADDLNK solo puede utilizarse para crear un enlace simbólico *con* un objeto de QSYS.LIB.
- Las operaciones de archivos solo 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.
- El mandato RCLLNK no está soportado.

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado "Acceso a los menús y las pantallas" en la página 72.

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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QSYS.LIB.

Las API listadas en el apartado "Realizar operaciones utilizando interfaces API" en la página 118 pueden realizar operaciones en el sistema de archivos QSYS.LIB, con las siguientes restricciones:

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

Información relacionada

Interfaces de programación de aplicaciones (API)

QSYS.LIB de ASP independiente

El sistema de archivos QSYS.LIB de ASP independiente ofrece soporte para la estructura de bibliotecas del servidor iSeries 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 servidor iSeries 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 servidor iSeries 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 servidor iSeries
- Soporta operaciones de E/S continuas en miembros de archivos físicos, espacios de usuario y archivos de salvar.

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

Es posible acceder a un sistema de archivos QSYS.LIB de ASP independiente a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

En lo referente al uso de las interfaces del sistema de archivos integrado, deberá tener en cuenta las consideraciones y limitaciones siguientes.

Referencia relacionada

"Sistema de archivos definido por el usuario y agrupaciones de almacenamiento auxiliar (ASP) independientes" en la página 44
Cuando se activa una ASP independiente, se producen diversos cambios en el sistema de archivos "raíz" (/).

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

A continuación, se proporcionan las restricciones que se deben tener en cuenta cuando se manejan archivos en el sistema 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().
- Si algún trabajo tiene abierto un miembro de archivo de base de datos, solo un trabajo tiene acceso de escritura a ese miembro de archivo en todo momento. Las demás peticiones solo tienen acceso de lectura.

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

QSYS.LIB de ASP independientes soporta operaciones de entrada y salida continuas en objetos espacio de usuario.

Por ejemplo, un programa puede escribir datos continuos en un espacio de usuario y leer datos en un espacio de usuario. El tamaño máximo de un espacio de usuario es 16.776.704 de 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 para salvar archivos en el sistema de archivos QSYS.LIB de ASP independiente

EL QSYS.LIB de ASP independiente admite las operaciones de E/S continuas para salvar 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 escritura, no se permiten otras instancias abiertas del archivo. Un archivo de salvar **permite** que existan varias instancias abiertas para lectura, a condición de que ningún trabajo tenga más de una instancia abierta del archivo para lectura. Un archivo de salvar no puede abrirse para acceso de lectura/escritura. Las operaciones de E/S continuas para salvar 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 vías 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 inequívocamente el 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, solo 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 el tema dedicado a la Globalización de i5/OS del iSeries Information Center.

Conceptos relacionados

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

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. Por lo tanto, 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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos mandatos y pantallas del sistema de archivos integrado son válidos en el sistema de archivos QSYS.LIB de ASP independiente.

Prácticamente todos los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QSYS.LIB de ASP independiente. No obstante, existen algunas excepciones:

- El mandato ADDLNK solo se puede utilizar para crear un enlace simbólico con un objeto de QSYS.LIB de ASP independiente.
- Las operaciones de archivos solo 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 salvar 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.
- El mandato RCLLNK no está soportado.

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado "Acceso a los menús y las pantallas" en la página 72.

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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QSYS.LIB de ASP independiente.

Las API que aparecen en el apartado “Realizar operaciones utilizando interfaces API” en la página 118 pueden realizar operaciones en el sistema de archivos QSYS.LIB de ASP independiente, excepto en los casos siguientes:

- Las operaciones de archivos solo se pueden efectuar con archivos físicos descritos por programa y con archivos físicos fuente.
- La función symlink() solo se puede utilizar 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.
- Si utiliza las API QsrSave() o QsrRestore() para salvar o restaurar objetos de biblioteca en una ASP independiente, esta ASP independiente debe asociarse con el trabajo que realiza la operación de salvar o restaurar, o es preciso especificar la ASP independiente en la clave ASPDEV. El convenio de denominación del nombre de vía de acceso (/asp_name/QSYS.LIB/object.type) no está soportado en las API QsrSave() y QsrRestore().

Información relacionada

Interfaces de programación de aplicaciones (API)

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.

Utilización de QDLS mediante la interfaz del sistema de archivos integrado

Es posible acceder a un sistema de archivos QDLS a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

En lo referente al uso de las interfaces del sistema de archivos integrado, deberá 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 solo a través de los mandatos CL de los objetos de biblioteca de documentos (DLO), sino también a través de la interfaz del sistema de archivos integrado o las API que proporciona un sistema de archivos jerarquizado (HFS).

El sistema de archivos integrado se basa en el modelo de programas Integrated Language Environment (ILE), mientras que el HFS se basa en el modelo de programas 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.

Información relacionada

Las API del sistema de archivos jerárquico

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 solo 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.

Información relacionada

Nombre de documento y carpeta

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.

- Por ejemplo:

  ```
  /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. Un nombre que no se pueda convertir a la página de códigos pertinente podría quedar rechazado.

  Para obtener más información sobre páginas de códigos, consulte el tema Globalización de i5/OS, en iSeries Information Center.

  Conceptos relacionados

  "Nombre de vía de acceso" en la página 16
  Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos de los mandatos y las pantallas del sistema de archivos integrado son válidos en el sistema de archivos QDLS.

Los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QDLS, excepto los siguientes mandatos:

- El mandato ADDLNK solo 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.
- Estos mandatos no están soportados:
  - APYJRNCHG
  - CHGJRNOBJ
  - DSPJRN
  - ENDJRN
  - RCLLNK
  - RCVJRNE
  - RTVJRNE
  - SNDJRNE
  - STRJRN

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado "Acceso a los menús y las pantallas" en la página 72.

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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QDLS.

Las API que aparecen en el apartado "Realizar operaciones utilizando interfaces API" en la página 118 pueden realizar operaciones en el sistema de archivos QDLS, excepto las siguientes API:

- La función symlink() solo puede utilizarse para enlazar con un objeto de QDLS desde otro sistema de archivos que ofrece soporte a enlaces simbólicos.
- Las funciones siguientes no están soportadas:
  - givedescriptor()
  - ioctl()
  - link()
  - QjoEndJournal()
  - QjoRetrieveJournalEntries()
  - QjoRetrieveJournalInformation()
  - QJORJIDI()
  - QJOSJRNE()
  - QjoStartJournal()
  - Qp0lGetPathFromFileID()
  - readlink()
  - takedescriptor()

Información relacionada

Interfaces de programación de aplicaciones (API)

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, por ejemplo, DOS y OS/2.
- Está optimizado para la entrada y salida de archivos continuos.
- Soporta datos almacenados en archivos continuos.

Utilización de QOPT en la interfaz del sistema de archivos integrado

Al sistema de archivos QOPT se puede acceder a través del sistema de archivos integrado utilizando las pantallas de usuario, las API y los mandatos del servidor de PC o 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.

Información relacionada

Programación de dispositivos ópticos

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 (HFS).

El sistema de archivos integrado se basa en el modelo de programas Integrated Language Environment (ILE), mientras que el HFS se basa en el modelo de programas 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 del HFS, consulte la publicación Optical device programming


.

Información relacionada

Las API del sistema de archivos jerárquico

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 /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 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.

Para obtener más información sobre las reglas de nombres de las vías de acceso en el sistema de archivos QOPT, consulte la descripción de las "Reglas de nombres de las vías de acceso" en la publicación Optical device programming  .

Conceptos relacionados

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Enlaces en el sistema de archivos QOPT

El sistema de archivos QOPT da soporte solo a 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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos de los mandatos y las pantallas del sistema de archivos integrado son válidos en el sistema de archivos QOPT.

La mayoría de los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QOPT. Sin embargo, hay algunas excepciones en el sistema de archivos QOPT. 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 "Acceso a los menús y las pantallas" en la página 72.

Los siguientes mandatos del sistema de archivos integrado no están soportados por el sistema de archivos QOPT:
- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHKIN
- CHKOUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- SNDJRNE
- STRJRN

- WRKOBJOWN
- WRKOBJPGP

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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QOPT.

Todas las API que figuran en el apartado "Realizar operaciones utilizando interfaces API" en la página 118 pueden operar en el sistema de archivos "raíz" (/) de un modo seguro en ejecución multihebra, excepto las siguientes:

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

Información relacionada

Interfaces de programación de aplicaciones (API)

Sistema de archivos NetWare (QNetWare)

El sistema de archivos QNetWare proporciona acceso a los datos en servidores de PC autónomos que ejecutan Novell NetWare 5.1 o 6.0.

Asimismo, QNetWare también ofrece la siguiente funcionalidad:

- 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.

Notas:

1. El sistema de archivos QNetWare está disponible solo si en el sistema está instalada la Integración mejorada de NetWare, 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.
2. El producto NetWare Enhanced Integration no da soporte a Novell Storage Services (NSS), por lo que el acceso a los datos contenidos en esa partición estará limitado o restringido.

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`.

El montaje de sistemas de archivos NetWare también se puede utilizar para sacar partido de las opciones del mandato Añadir sistema de archivos montado (ADDMFS); por ejemplo, montar un sistema de archivos de lectura-escritura como solo de lectura. Consulte el tema Mandato Añadir sistema de archivos montado (ADDMFS) para obtener más información.

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
  ```

  La extensión `.TRE` se utiliza para representar árboles NDS, `.C` representa a países, `.O` representa a organizaciones, `.OU` representa a unidades organizativas y `.CN` se utiliza para nombre propios. 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 en la interfaz del sistema de archivos integrado

Es posible acceder a un sistema de archivos QNetWare a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS 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 a 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, 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, las autorizaciones 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 Cambiar valor de auditoría (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 solo 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 de los servicios de directorio NetWare en el sistema de archivos QNetWare

El sistema de archivos QNetWare da soporte a la visualización de nombres de los servicios de directorio NetWare (NDS) en mayúsculas y en minúsculas.

Enlaces en el sistema de archivos QNetWare

El sistema de archivos QNetWare da soporte solo 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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos de los mandatos y las pantallas del sistema de archivos integrado son válidos en el sistema de archivos QNetWare.

Los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QNetWare, con las siguientes excepciones:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGJRNOBJ
- CHGPGP
- CHKIN
- CHKOUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- SNDJRNE
- 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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QNetWare.

Las API que aparecen en el apartado "Realizar operaciones utilizando interfaces API" en la página 118 pueden realizar operaciones en el sistema de archivos QNetWare, excepto las siguientes API:

- givedescriptor()
- link()
- QjoEndJournal()
- QjoRetrieveJournalEntries()
- 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()
- writev()

Información relacionada

Interfaces de programación de aplicaciones (API)

Sistema de archivos iSeries NetClient (QNTC)

El sistema de archivos QNTC proporciona acceso a los datos y objetos que se encuentran almacenados en un Integrated xSeries Server para iSeries que ejecuta Windows NT 4.0 Server o superior, o Linux. El sistema de archivos QNTC proporciona acceso a los datos y objetos que se encuentran almacenados en servidores remotos que ejecutan Windows NT 4.0 o posterior, Linux Samba 3.0 o posterior, o versiones soportadas de iSeries NetServer.

El sistema de archivos QNTC permite a las aplicaciones de iSeries utilizar los datos almacenados en servidores Windows o Linux.

El sistema de archivos QNTC forma parte del sistema operativo i5/OS base. Para acceder a /QNTC, no es necesario tener instalada la opción 29 del sistema operativo, Integrated Server Support.

Utilización de QNTC mediante la interfaz del sistema de archivos integrado

Al utilizar los mandatos, las pantallas de usuario y las API de iSeries NetServer, iSeries Navigator o el sistema de archivos integrado, puede acceder al sistema de archivos QNTC a través de la interfaz del sistema de archivos integrado.

Tenga en cuenta las siguientes consideraciones y limitaciones.

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

| La vía de acceso está formada por el nombre de sistema de archivos, el nombre del servidor, el nombre
| compartido, los nombres del directorio y el subdirectorio, y el nombre del objeto.

Los requisitos de un nombre de vía de acceso son los siguientes:

- 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 tienen el formato siguiente:

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

- Los nombres de vía de acceso son sensibles a las mayúsculas y minúsculas.
- 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.
- | Por omisión, cada servidor soportado en funcionamiento en la subred local aparecerá automáticamente
| como un directorio bajo /QNTC. Utilice el mandato Crear directorio (MKDIR) o la API mkdir() para
| añadir servidores accesibles ubicados fuera de la subred local.

Conceptos relacionados

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Información relacionada

Mandato Crear directorio (MKDIR)

API Crear directorio (mkdir())

Enlaces en el sistema de archivos QNTC

El sistema de archivos QNTC da soporte solo 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.

Conceptos relacionados

"Enlace" en la página 12
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.

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

Muchos de los mandatos y las pantallas del sistema de archivos integrado son válidos en el sistema de archivos QNTC.

Los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QNTC, excepto los siguientes mandatos:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHGOWN
- CHGAUT
- CHGPGP
- CHKIN
- CHKOUT
- DSPAUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- RST (disponible con Servidores integrados xSeries)
- SAV (disponible con Servidores integrados xSeries)
- SNDJRNE
- STRJRN
- WRKAUT
- WRKOBJOWN
- WRKOBJPGP

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado "Acceso a los menús y las pantallas" en la página 72.

Variables de entorno de QNTC

El comportamiento de QNTC al examinar la red se puede controlar mediante dos variables de entorno. El soporte de estas variables de entorno empezó en i5/OS V5R4. Utilice el mandato CL ADDENVVAR para crear estas variables de entorno.

QZLC_SERVERLIST

Cuando esta variable de entorno se establece en ″2″, QNTC puede acceder a todos los servidores que aparecen en el directorio /QNTC en el sistema de archivos integrado. Este era el comportamiento por omisión antes de V5R4. Cuando esta variable no se establece en ″2″ o no se ha creado, puede que no se pueda acceder a algunos de los servidores que aparecen en el directorio /QNTC.

QIBM_ZLC_NO_BROWSE

Cuando esta variable de entorno se establece en ″1″, el directorio /QNTC solo contendrá servidores que se han creado con el mandato CL MKDIR o la API mkdir(). El rendimiento de muchas operaciones en el sistema de archivos QNTC aumenta cuando se ha establecido esta variable de entorno. No obstante, es necesario crear todos los directorios /QNTC utilizando el mandato CL.

Crear directorios en el sistema de archivos QNTC

Utilice el mandato Crear directorio (MKDIR) o la API mkdir() para añadir un directorio de servidor al directorio /QNTC.

Por omisión, se crea automáticamente un directorio QNTC para todos los servidores en funcionamiento en el dominio iSeries NetServer y la subred local. Los servidores externos a la subred local o el dominio iSeries NetServer 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:

- Configurando iSeries NetServer para WINS, es posible crear automáticamente directorios para servidores externos a la subred.
- Si utiliza la API mkdir() o el mandato CL MKDIR para añadir directorios a la estructura de directorios, estos 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.

Si prefiere añadir directorios utilizando la API o el mandato CL, puede aumentar el rendimiento de estos mandatos añadiendo la variable de entorno QIBM_ZLC_NO_BROWSE, tal como se muestra en el siguiente ejemplo:

```
ADDENVVAR ENVVAR(QIBM_ZLC_NO_BROWSE) VALUE(1) LEVEL(*SYS)
```

Esta variable de entorno hace que el sistema de archivos omita todas las operaciones de examinar la red cuando se realizan operaciones de archivos.

Información relacionada

Mandato Crear directorio (MKDIR)

API Crear directorio (mkdir())

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

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QNTC.

Las API que aparecen en el apartado “Realizar operaciones utilizando interfaces API” en la página 118 pueden realizar operaciones en el sistema de archivos QNTC, excepto las siguientes API:

- 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()
  - fclear()
  - fclear64()
  - givedescriptor()
  - link()
  - QjoEndJournal()
  - QjoRetrieveJournalEntries()
  - QjoRetrieveJournalInformation()
  - QJORJIDI()
  - QJOSJRNE()
  - QjoStartJournal()
  - Qp0lGetPathFromFileID()
  - readlink()
  - symlink()
  - takedescriptor()

Información relacionada

Interfaces de programación de aplicaciones (API)

Habilitar el sistema de archivos QNTC para el Servicio de autenticación de red

QNTC permite a iSeries acceder a servidores CIFS que dan soporte al protocolo de autenticación Kerberos V5.

En lugar de utilizar una contraseña tipo administrador LAN para la autenticación con cada servidor, un servidor iSeries correctamente configurado podrá acceder a servidores CIFS soportados con una sola transacción de inicio de sesión.

Para habilitar el Servicio de autenticación de red (NAS) para utilizarlo con QNTC, debe configurar los elementos:

- Servicio de autenticación de red (NAS)
- Enterprise Identity Mapping (EIM)

Tras configurar los elementos anteriores, un usuario podrá utilizar NAS con el sistema de archivos QNTC. Para que un usuario pueda aprovechar el soporte de NAS de QNTC, deberá llevar a cabo los pasos siguientes.

- El perfil de usuario de iSeries para ese usuario debe tener el parámetro de gestión de contraseñas local LCLPWDMGT establecido en ***NO**. Especificando *NO, el usuario no tendrá una contraseña para el sistema y no podrá iniciar una sesión 5250. El único acceso al sistema será a través de aplicaciones habilitadas para NAS, como iSeries Navigator o iSeries Access 5250 Display Emulator.

  Si el usuario especifica *YES, el sistema gestionará la contraseña y el usuario se autenticará sin NAS.
- Debe tener un ticket kerberos y una conexión de iSeries Navigator.
- El ticket kerberos para el iSeries que está utilizando debe poder enviarse. Para poder enviar un ticket, siga estos pasos:

– Acceda a la herramienta ″Usuarios y ordenadores del directorio activo″ en el KDC para su reino NAS
– Seccione los usuarios
– Seleccione el nombre que corresponde al nombre principal del servicio
– Seleccione Propiedades
– Seleccione la pestaña Cuenta
– En las opciones de Cuenta, marque ″Se confía en la cuenta para delegación″

Información relacionada

El servicio de autenticación de red

Enterprise Identity Mapping (EIM)

Sistema de archivos del servidor de archivos i5/OS (QFileSvr.400)

El sistema de archivos QFileSvr.400 proporciona acceso transparente a otros sistemas de archivos que residen en servidores iSeries remotos. Se accede a él 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 interacciona con el servidor de archivos i5/OS del sistema de destino para efectuar la operación de archivo real.

Utilización de QFileSvr.400 en la interfaz del sistema de archivos integrado

Es posible acceder a un sistema de archivos QFileSvr.400 a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos i5/OS o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado.

En lo referente al uso de las interfaces del sistema de archivos integrado, deberá 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 QFileSvr.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 i5/OS.

Nombres de vías de acceso en el sistema de archivos QFileSvr.400

Los nombres de vías de acceso tiene un forma específico en el sistema de archivos QFileSvr.400.

- El formato es:

  ```
  /QFileSvr.400/RemoteLocationName/Directory/Directory . . . /Object
  ```

  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.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Comunicaciones en el sistema de archivos QFileSvr.400

El sistema de archivos QFileSvr.400 se comunica de las siguientes formas.

- Las conexiones TCP con el servidor de archivos de un servidor de destino pueden establecerse solo si el subsistema QSERVER del servidor de destino está activo.
- Las conexiones LU 6.2 SNA solo 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 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 i5/OS 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 i5/OS 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 i5/OS 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 i5/OS

- El sistema de archivos QFileSvr.400 efectúa una comprobación periódica (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/Remote2/QFileSvr.400/Remote1/QFileSvr.400/Remote2/...
  ```

  donde `Remote1` 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 QFileSvr.400

Si ambos sistemas tienen configurado el Servicio de autenticación de red y Enterprise Identity Mapping (EIM), y el usuario se ha autenticado con Kerberos, puede utilizarse Kerberos para autenticarse para acceder a 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 acceder a un sistema de archivos que resida en un servidor iSeries 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, solo se puede acceder 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.

Información relacionada

El servicio de autenticación de red

Enterprise Identity Mapping (EIM)

Enlaces en el sistema de archivos QFileSvr.400

El sistema de archivos QFileSvr.400 da soporte solo a 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.

Conceptos relacionados

"Enlace" en la página 12
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.

Utilización de los mandatos y las pantallas del sistema de archivos integrado en el sistema de archivos QFileSvr.400

Muchos de los mandatos y las pantallas del sistema de archivos integrado son válidos en el sistema de archivos QFileSvr.400.

Los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 pueden realizar operaciones en el sistema de archivos QFileSvr.400, excepto los siguientes mandatos:

- ADDLNK
- APYJRNCHG
- CHGAUT
- CHGJRNOBJ
- CHGOWN
- DSPAUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RST
- RTVJRNE
- SAV
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado "Acceso a los menús y las pantallas" en la página 72.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QFileSvr.400

Muchas de las API del sistema de archivos integrado son válidas en el sistema de archivos QFileSvr.400.

Las API que aparecen en el apartado "Realizar operaciones utilizando interfaces API" en la página 118 pueden realizar operaciones en el sistema de archivos QFileSvr.400, excepto las siguientes API:

- chown()
- fchown()
- fclear()
- fclear64()
- givedescriptor()
- link()

- QjoEndJournal()
- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE
- QjoStartJournal
- Qp0lGetPathFromFileID()
- symlink()
- takedescriptor()

Información relacionada

Interfaces de programación de aplicaciones (API)

Sistema de archivos de red (NFS)

El sistema de archivos de red (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.

Utilización de NFS mediante la interfaz del sistema de archivos integrado

Puede acceder al NFS a través de la interfaz del sistema de archivos integrado. Tenga en cuenta estas 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 (Network File System (NFS)) y sus características.

Las tres posibilidades son:

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 en un cliente de un servidor que no es un servidor iSeries ni un servidor UNIX.

En el primer caso, el sistema de archivos montado se comportará en el cliente exactamente igual que en el servidor iSeries. 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.

En el tercer caso, será necesario leer la documentación del sistema de archivos asociado con el sistema operativo del servidor.

Referencia relacionada

"Sistema de archivos de sistemas abiertos (QOpenSys)" en la página 36
El sistema de archivos QOpenSys es compatible con los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX y X/Open Portability Guide (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.

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.

Conceptos relacionados

"Enlace" en la página 12
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.

Mandatos del sistema de archivos integrado en el sistema de archivos de red

Muchos mandatos del sistema de archivos integrado son válidos en el sistema de archivos de red (NFS)

Todos los mandatos listados en el apartado "Acceso mediante mandatos CL" en la página 73 y las pantallas descritas en el apartado "Acceso a los menús y las pantallas" en la página 72 pueden realizar operaciones en el sistema de archivos de red, excepto los siguientes mandatos:

- APYJRNCHG
- CHGJRNOBJ
- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP

- CHKIN
- CHKOUT
- DSPJRN
- ENDJRN
- RCLLNK
- RCVJRNE
- RTVJRNE
- 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 una descripción completa de los mandatos y pantallas relacionados específicamente con el sistema de archivos de red en OS/400 Network File System Support .

Tabla 6. Mandatos CL del sistema de archivos de red

Mandato	Descripción
ADDMFS	Añadir sistema de archivos montado. Pone los sistemas de archivos de servidor remoto exportados en directorios de cliente locales.
CHGNFSEXP	Cambiar sistema de archivos de red. Exportar. 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 un 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 un sistema de archivos. Pone los sistemas de archivos de servidor remoto exportados en directorios de cliente locales. Es un alias del mandato ADDMFS.
RLSIFSLCK	Liberar bloqueos de sistema de archivos integrado. Libera todos los bloqueos de rango de byte de Network mantenidos por un cliente o sobre un objeto.
RMVMFS	Eliminar sistema de archivos montado. Elimina los sistemas de archivos de servidor remoto exportados de los espacios de nombre de cliente locales.
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 un sistema de archivos. Elimina los sistemas de archivos de servidor remoto exportados de los espacios de nombre de cliente locales. Es un alias del mandato RMVMFS.

Nota: para poder utilizar mandatos en un sistema de archivos de red, primero hay que montarlo.

Interfaces API de sistema de archivos integrado en el sistema de archivos de red

Muchas API del sistema de archivos integrado son válidas en el sistema de archivos de red (NFS)

Todas las API que se listan en el apartado “Realizar operaciones utilizando interfaces API” en la página 118 pueden realizar operaciones en el sistema de archivos de red, excepto las siguientes:

- QjoEndJournal()

- QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Encontrará una descripción completa de las funciones de 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 hay que montarlo.

Información relacionada

Interfaces de programación de aplicaciones (API)

Acceso al sistema de archivos integrado

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 interaccionar con los directorios y objetos del sistema de archivos integrado desde el servidor por medio de menús y pantallas o mediante mandatos de 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 interaccionar con el sistema de archivos integrado mediante iSeries Navigator, una interfaz gráfica de usuario utilizada para gestionar y administrar el servidor desde el escritorio Windows.

Acceso a los menús y las pantallas

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. En el menú principal de iSeries, seleccione la opción **Archivos, bibliotecas y carpetas**.
4. En el menú Archivos, bibliotecas y carpetas, seleccione la opción **Sistema de archivos integrado**.

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 **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 o mandatos 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 redenominar objetos
- Reservar y reincorporar objetos
- Salvar y restaurar objetos
- Visualizar y cambiar propietarios de objetos y autorizaciones de usuarios
- Visualizar y cambiar los atributos de los objetos
- 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.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

Referencia relacionada

"Reglas que rigen los nombres de vía de acceso en los mandatos CL y pantallas" en la página 77
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.

"Acceso mediante mandatos CL"
Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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.

"Utilización de mandatos del sistema de archivos integrado en el sistema de archivos "raíz" (/)" en la página 35
Todos los mandatos listados en el tema Acceso mediante mandatos CL y las pantallas que se describen en el tema Acceso a los menús y las pantallas pueden realizar operaciones en el sistema de archivos "raíz" (/). Sin embargo, puede que la ejecución de estos mandatos en un proceso que admite la ejecución multihebra no sea segura.

Acceso mediante mandatos CL

Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado, también pueden efectuarse entrando mandatos de 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 “Sistemas de archivos definidos por el usuario (UDFS)” en la página 38 y “Sistema de archivos de red (NFS)” en la página 69. 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 7. 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 producidos desde que se salvó 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
CHGJRNOBJ[2]	Cambiar objetos registrados por diario. Cambia los atributos de registro por diario de un objeto o lista de objetos sin necesidad de finalizar y reiniciar el registro por diario para el objeto.	
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 desde archivo continuo. Copia los datos de un archivo continuo en un miembro de archivo de base de datos.	
CPYTOSTMF	Copiar en 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 archivos integrado del formato *TYPE1 al formato *TYPE2.	

Tabla 7. Mandatos del sistema de archivos integrado (continuación)

Mandato	Descripción	Alias
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.	
DSPJRN[2]	Visualizar diario. Convierte las entradas de diario (contenidas en uno o varios receptores) en un formato adecuado para la representación externa.	
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
PRTDIRINF	Imprimir información de directorio. Se utiliza para imprimir la información de directorio para objetos del sistema de archivos integrado recogido mediante el mandato Recuperar información de directorio (RTVDIRINF).	
RCLLNK	Reclamar enlaces de objetos. Identifica y, si es posible, corrige los problemas en los sistemas de archivos montados que se están utilizando.	
RCVJRNE[2]	Recibir entrada de diario. Permite que un programa de salida de usuario especificado reciba continuamente entradas de diario.	
RLSIFSLCK	Liberar bloqueos de sistema de archivos integrado. Libera todos los bloqueos de rango de byte mantenidos por un cliente NFS 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

Tabla 7. Mandatos del sistema de archivos integrado (continuación)

Mandato	Descripción	Alias
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).	
RTVDIRINF	Recuperar información de directorio. Se utiliza para recoger los atributos de objetos del sistema de archivos integrado.	
RTVJRNE[2]	Recuperar entrada de diario. Permite obtener una determinada entrada de diario y coloca los resultados en variables CL.	
SAV	Salvar. Copia un objeto o un grupo de objetos del sistema en 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.	
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.	

Notas:

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 en el Information Center de iSeries para obtener más información.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

Tareas relacionadas

"Acceso a los menús y las pantallas" en la página 72
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.

Referencia relacionada

"Utilización de mandatos del sistema de archivos integrado en el sistema de archivos ″raíz″ (/)" en la página 35
Todos los mandatos listados en el tema Acceso mediante mandatos CL y las pantallas que se describen en el tema Acceso a los menús y las pantallas pueden realizar operaciones en el sistema de archivos "raíz" (/). Sin embargo, puede que la ejecución de estos mandatos en un proceso que admite la ejecución multihebra no sea segura.

Información relacionada

Lenguaje de control (CL)

Reglas que rigen los nombres de vía de acceso en los 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.

La lista siguiente es una visión general 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 cualquier directorio, archivo, enlace u 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 ir entre comillas simples ('). Cuando los nombres de vías de acceso se escriben en las pantallas, las comillas son opcionales. No obstante, si el nombre de vía de acceso contiene series entrecomilladas, 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 (/).

  Nota: Algunos mandatos CL también permiten utilizar la barra inclinada invertida (\) como separador, convirtiendo automáticamente la barra inclinada invertida (\) en una barra inclinada (/). No obstante, otros mandatos CL tratan la barra inclinada invertida (\) de la misma forma que cualquier otro carácter. Por lo tanto, el separador de barra inclinada invertida (\) debe utilizarse con precaución.

  Por ejemplo:

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

  o

  ```
  'Dir1\Dir2\Dir3\ArchUsr'
  ```
- Los caracteres de barra inclinada (/) y barra inclinada invertida (\) y los nulos no se pueden utilizar en los componentes individuales del nombre de vía de acceso cuando la barra inclinada (/) y la barra inclinada invertida (\) 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.
- Un carácter separador (por ejemplo: /) 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 separador (por ejemplo: /), 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 un carácter separador (por ejemplo: /) 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 un carácter separador (por ejemplo: /) 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:

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

  Nota: este caso solo atañe a los nombres de vía de acceso relativa, en los que no hay otros caracteres antes del asterisco (*).
- Cuando se trabaja con objetos del sistema de archivos QSYS.LIB, los nombres de componentes deben tener el formato *nombre.tipo-objeto*; por ejemplo:

  ```
  '/QSYS.LIB/PAY.LIB/TAX.FILE'
  ```
- Cuando se trabaja con objetos del sistema de archivos QSYS.LIB de ASP independiente, los nombres de componentes deben tener el formato *nombre.tipo-objeto*; por ejemplo:

  ```
  '/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE
  ```
- El nombre de vía de acceso debe estar delimitado por conjuntos adicionales de comillas simples (') o comillas dobles (") si se utiliza alguno de los caracteres siguientes en un nombre de componente:
  - Asterisco (*)

    Nota: Para evitar confusiones con los valores especiales del servidor iSeries, los nombres de vía de acceso no deben comenzar con un único carácter de asterisco (*).
  - Interrogante (?)
  - Comillas simples (')
  - Comillas dobles (")
  - 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*Smith"'
  ```

  o

  ```
  '''/Dir1/Dir/A*Smith'''
  ```

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 solo atañe a mandatos y pantallas asociadas, no a las API. Además, no se permite el valor 0 hexadecimal en los nombres de las vías de acceso.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Referencia relacionada

"Reglas de nombres de vía de acceso para las API" en la página 126
Cuando se utiliza una API del sistema de archivos integrado o de ILE C/400 para realizar operaciones sobre un objeto, este se identifica especificando su vía de acceso de directorio. A continuación se proporciona una visión general de las reglas que hay que tener en cuenta al especificar nombres de vías de acceso en las API.

Información relacionada

Lenguaje de control (CL)

Trabajar con los resultados de los mandatos RTVDIRINF y PRTDIRINF

El mandato Recuperar información de directorios (RTVDIRINF) se utiliza para recoger los atributos de objetos del sistema de archivos integrado. La información recogida se almacena en archivos de bases de datos (tablas) a los que se les asignan nombres utilizando el prefijo del archivo de información especificado por el parámetro INFILEPFX. Las tablas se crean en la biblioteca especificada por el parámetro INFLIB.

Como resultado del mandato RTVDIRINF se crean tres tablas. Una tabla almacena atributos de objeto, otra es para los directorios y la última se utiliza para determinar qué archivos se han utilizado para almacenar los atributos de los objetos.

En la tabla siguiente se describen los campos facilitados para la tabla que almacena los atributos del objeto. Si se especifica *GEN en el parámetro de prefijo del archivo de información (INFFILEPFX), los archivos de base de datos se crean con un prefijo único generado por este mandato. El prefijo empieza por QAEZD seguido de cuatro dígitos. A los archivos creados para almacenar la información recogida se les asignan nombres utilizando este prefijo seguido por la letra D (para el archivo que contiene información del directorio) o por la letra O (para el archivo que contiene información sobre los objetos del directorio). Por ejemplo, la primera vez que se ejecute el mandato especificando *GEN, se crean los archivos QAEZD0001D y QAEZD0001O en la biblioteca especificada por el parámetro de biblioteca de información (INFLIB). Los usuarios pueden especificar un prefijo de archivo que se utilizará para asignar un nombre a esta base de datos, que puede tener una extensión máxima de nueve caracteres.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto)

Nombre del campo	Tipo de campo	Descripción del campo
QEZACCTIM	TIMESTAMP	La fecha y hora en las que se accedió por última vez a los datos del objeto.
QEZALCSIZE [1]	BIGINT	El número de bytes asignados para este objeto.
QEZALWCKPW	SMALLINT	Indica si un archivo continuo (*STMF) puede compartirse con lectores y transcriptores durante el proceso de punto de control de salvar mientras está activo. Los valores válidos son: 0 - El objeto puede compartirse solo con lectores. 1 - El objeto puede compartirse con lectores y transcriptores.
QEZASP	SMALLINT	La agrupación de almacenamiento auxiliar en la que se almacena el objeto.
QEZAUDT	GRAPHIC(10)	El valor de auditoría asociado con el objeto. Los valores válidos son: *NONE - No se producirá ninguna auditoría para este objeto cuando se lea o modifique, sea cual sea el usuario que acceda al objeto. *USRPRF - Solo se efectuará una auditoría de este objeto si se está auditando al usuario actual. Se analiza al usuario actual para determinar si debería efectuarse una auditoría para este objeto. El perfil de usuario puede especificar si solo se audita el acceso de modificación o si se auditan los accesos de lectura y modificación para este objeto. *CHANGE - Se efectuarán auditorías de todos los accesos de modificación a este objeto por parte de todos usuarios del sistema. *ALL - Se efectuarán auditorías de todos los accesos a este objeto por parte de todos usuarios del sistema. Todo acceso se define como operación de lectura o modificación.
QEZAUTLST	GRAPHIC(10)	El nombre de la lista de autorización que se utiliza para asegurar el objeto mencionado. El valor *NONE indica que no se utiliza ninguna lista de autorización al determinar la autorización para el objeto.
QEZBLKSIZ	INTEGER	El tamaño de bloque de un objeto.
QEZCASE	SMALLINT	Indica la distinción entre mayúsculas/minúsculas del sistema de archivos que contiene este objeto. 0 - El sistema de archivos no distingue mayúsculas/minúsculas. 1 - El sistema de archivos distingue mayúsculas/minúsculas.
QEZCCSID	INTEGER	El CCSID de los datos y atributos ampliados del objeto.
QEZCEAS	BIGINT	Número de atributos ampliados críticos asociados con este objeto.
QEZCHGTIMA [1]	TIMESTAMP	La fecha y hora en las que se cambiaron por última vez los atributos del objeto.
QEZCHGTIMD	TIMESTAMP	La fecha y hora en las que se cambiaron por última vez los datos del objeto.
QEZCHKOUT [1]	SMALLINT	Un indicador sobre si se ha reservado un objeto. Los valores válidos son: 0 - El objeto no se ha reservado. 1 - El objeto se ha reservado.
QEZCHKOWN	GRAPHIC(10)	El usuario que ha reservado el objeto. Este campo aparece en blanco si no se ha reservado.
QEZCHKTIM	TIMESTAMP	La fecha y hora en las que se reservó el objeto. Si no se ha reservado el objeto, el valor de este campo será NULL.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZCLSTRSP	SMALLINT	El objeto es el almacenamiento asignado para los servidores xSeries integrados para su utilización como unidades de disco virtual para los servidores xSeries. Desde la perspectiva del servidor iSeries, las unidades virtuales aparecen como archivos continuos en el sistema de archivos integrado. 0 - El objeto no es un almacenamiento en disco virtual. 1 - El objeto es un almacenamiento en disco virtual.
QEZCRTAUD	GRAPHIC(10)	El valor de auditoría asociado con un objeto creado en este directorio. Los valores válidos son: *NONE - No se producirá ninguna auditoría para este objeto cuando se lea o modifique, sea cual sea el usuario que acceda al objeto. *USRPRF - Solo se efectuará una auditoría de este objeto si se está auditando al usuario actual. Se analiza al usuario actual para determinar si debería efectuarse una auditoría para este objeto. El perfil de usuario puede especificar si solo se auditan los accesos de modificación o si se auditan los accesos de lectura y modificación para este objeto. *CHANGE - Se efectuarán auditorías de todos los accesos de modificación a este objeto por parte de todos usuarios del sistema. *ALL - Se efectuarán auditorías de todos los accesos a este objeto por parte de todos usuarios del sistema. Todos los accesos se definen como operación de lectura o modificación.
QEZCRTTIM	TIMESTAMP	La fecha y hora en las que se creó el objeto.
QEZDIRIDX	INTEGER	El índice del directorio padre.
QEZDIRTYP2	SMALLINT	El formato del objeto de directorio especificado. Los valores válidos son: 0 - El formato del directorio es *TYPE1. 1 - El formato del directorio es *TYPE2.
QEZDOM	GRAPHIC(10)	El dominio del objeto. Los valores válidos son: *SYSTEM - El objeto existe en el dominio del sistema. *USER - El objeto existe en el dominio del usuario.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZDSTGOPT	SMALLINT	Esta opción debería utilizarse para determinar cómo el sistema asigna el almacenamiento auxiliar para el objeto especificado. Esta opción solo se puede especificar para archivos continuos en los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario. Esta opción se ignorará para los archivos continuos *TYPE1. Los valores válidos son: 0 - El almacenamiento auxiliar se asignará de la forma habitual. Es decir, cuando se necesite almacenamiento auxiliar adicional, se asignará en extensiones de un tamaño lógico según las necesidades actuales de espacio y anticipándose a las necesidades futuras, al mismo tiempo que se minimiza el número de operaciones de E/S en disco. 1 - El almacenamiento auxiliar se asignará para minimizar el espacio utilizado por el objeto. Es decir, cuando se necesite almacenamiento auxiliar adicional, este se asignará en extensiones de tamaño reducido según las necesidades de espacio actuales. Acceder a un objeto compuesto por muchas extensiones pequeñas puede incrementar el número de operaciones de E/S en disco para dicho objeto. 2 - El sistema determinará dinámicamente la asignación óptima de almacenamiento auxiliar para el objeto, equilibrando el espacio utilizado frente a las operaciones de E/S en disco. Por ejemplo, si un archivo tiene muchas extensiones reducidas pero se lee y escribe con frecuencia, las asignaciones futuras de almacenamiento auxiliar serán extensiones mayores para minimizar el número de operaciones de E/S en disco. Por otra parte, si un archivo se trunca con frecuencia, las asignaciones de almacenamiento auxiliar futuras serán extensiones pequeñas para minimizar el espacio utilizado. Adicionalmente, se actualizará la información referente a los tamaños de los archivos continuos para este sistema y su actividad. Esta información sobre el tamaño de los archivos también se utilizará para determinar las asignaciones de almacenamiento auxiliar óptimas para este objeto en relación con el resto de tamaños de objeto.
QEZDTASIZE	BIGINT	El tamaño en bytes de los datos en este objeto. Este tamaño no incluye cabeceras de objeto o el tamaño de atributos ampliados asociados con el objeto.
QEZEAS	BIGINT	Número de atributos ampliados asociados con este objeto.
QEZEXTATRS	BIGINT	Número total de bytes par todos los datos de atributos ampliados.
QEZFILEID [1]	GRAPHIC (16)	El ID del archivo para el objeto. Un identificador asociado con el objeto. Puede utilizarse el ID de un archivo con Qp0lGetPathFromFileID() para recuperar el nombre de la vía de acceso de un objeto.
QEZFILEIDS	INTEGER	El ID de archivo de 4 bytes del archivo. Este número identifica inequívocamente el objeto dentro de un sistema de archivos. Este número no puede identificar el objeto en todo el sistema.
QEZFILTYP2 [1]	SMALLINT	El formato del archivo continuo (*STMF). Los valores válidos son: 0 - El formato del archivo continuo es *TYPE1. 1 - El formato del archivo continuo es *TYPE2.
QEZFSID	BIGINT	El ID del sistema de archivos al que pertenece el objeto. Este número identifica inequívocamente el sistema de archivos al que pertenece el objeto.
QEZGENID	BIGINT	El ID de generación asociado con el ID del archivo.
QEZGID	INTEGER	Los perfiles de grupo se identifican mediante un número de identificación de grupo (GID) numérico y exclusivo.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZINHSCN	GRAPHIC (1)	Especifica si los objetos creados en un directorio se explorarán cuando haya programas de salida registrados en cualquiera de los puntos de salida relacionados con la exploración del sistema de archivos integrado. Los valores válidos son: x'00' - Tras crear un objeto en el directorio, el objeto no se explorará según las reglas descritas en los programas de salida relacionados con la exploración. **Nota:** si no se especifica el valor *NOPOSTRST en Control de exploración de sistemas de archivos (QSCANFSCTL) cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración. x'01' - El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración si el objeto ha sido modificado o si el software de exploración ha sido actualizado desde la última vez en que se exploró el objeto. x'02' - El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración si el objeto ha sido modificado o si el software de exploración ha sido actualizado desde la última vez en que se exploró el objeto. No se explorará si se ha actualizado el software de exploración. Este atributo solo entra en vigor si para el valor del sistema Control de exploración de sistemas de archivos (QSCANFSCTL) se ha especificado *USEOCOATR. De lo contrario, se considerará que el atributo es SCANNING_YES. **Nota:** si no se especifica el valor *NOPOSTRST en Control de exploración de sistemas de archivos (QSCANFSCTL) cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración.
QEZJAFTERI	SMALLINT	Cuando el registro por diario está activo, se registra por diario la imagen del objeto tras un cambio. 0 - El objeto no se registra por diario con las imágenes posteriores. 1 - El objeto se registra por diario con las imágenes posteriores.
QEZJBEFORI	SMALLINT	Cuando el registro por diario está activo, se registra por diario la imagen del objeto antes de un cambio. 0 - El objeto no se registra por diario con las imágenes anteriores. 1 - El objeto se registra por diario con las imágenes anteriores.
QEZJOPTENT	SMALLINT	Cuando el registro por diario está activo, se registran por diario las entradas que se consideran opcionales. La lista de entradas por diario opcionales varía para cada tipo de objeto. 0 - El objeto no se registra por diario con entradas opcionales. 1 - El objeto se registra por diario con entradas opcionales.
QEZJRCVASP	GRAPHIC(10)	El nombre de la ASP que contiene el receptor de diario necesario para aplicar satisfactoriamente los cambios de diario. Los valores válidos son: *SYSBAS - El receptor de diario reside en la ASP del usuario o el sistema. Dispositivo ASP - El nombre del dispositivo ASP que contiene el receptor de diario.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZJRCVLIB	GRAPHIC(10)	El nombre de la biblioteca que contiene el receptor de diario necesario para aplicar satisfactoriamente los cambios de diario. Este campo está en blanco si el objeto no se ha registrado nunca por diario.
QEZJRCVNAM	GRAPHIC(10)	El receptor de diario más antiguo necesario para aplicar satisfactoriamente los cambios de diario. Si el campo Aplicar información se establece en PARTIAL_TRANSACTION, el receptor de diario contiene el inicio de la transacción parcial. De lo contrario, el receptor de diario contiene el inicio de la operación de salvar. Este campo está en blanco si el objeto no se ha registrado nunca por diario.
QEZJRNID	GRAPHIC(10)	Este campo asocia el objeto que se está registrando por diario con un identificador que puede utilizarse en varios mandatos y API relacionados con el registro por diario. Este campo está en blanco si el objeto no se ha registrado nunca por diario.
QEZJRNLIB	GRAPHIC(10)	Si el valor del estado de registro por diario es JOURNALED, este campo contiene el nombre de la biblioteca que contiene el diario utilizado actualmente. Si el valor del estado de registro por diario es NOT_JOURNALED, este campo contiene el nombre de la biblioteca con el último diario utilizado. Este campo está en blanco si el objeto no se ha registrado nunca por diario.
QEZJRNNAM	GRAPHIC(10)	Si el valor del estado de registro por diario es JOURNALED, este campo contiene el nombre del diario que actualmente se utiliza. Si el valor del estado de registro por diario es NOT_JOURNALED, este campo contiene el nombre del último diario utilizado para este objeto. Este campo está en blanco si el objeto no se ha registrado nunca por diario.
QEZJRNSTR	TIMESTAMP	El número de segundos desde la Época que corresponde a la última fecha y hora en las que se inició el registro por diario del objeto por última vez. Este campo tiene un valor NULL si el objeto no se ha registrado nunca por diario.
QEZJRNSTS [1]	SMALLINT	El estado actual de registro por diario del objeto. Este campo tendrá uno de los valores siguientes: 0 (NOT_JOURNALED) - Actualmente, el objeto no está siendo registrado por diario. 1 (JOURNALED) - Actualmente, el objeto se está registrando por diario.
QEZJSUBTRE	SMALLINT	Cuando se devuelve este distintivo, este objeto es un directorio del sistema de archivos integrado con registro por diario con semántica de subárbol. 0 - El objeto no se registra por diario con semántica de subárbol. 1 - El objeto se registra por diario con semántica de subárbol. Los objetos nuevos que se creen dentro del subárbol de este directorio heredarán los atributos de registro por diario y las opciones de este directorio.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZJTRNI	GRAPHIC (1)	Este campo describe información sobre el estado actual del objeto en relación con los límites de control de compromiso. Los valores válidos son: x'00' (NONE) - No hay transacciones parciales. x'01' (PARTIAL_TRANSACTION) - El objeto se restauró con transacciones parciales. Este objeto no puede utilizarse hasta que se utilice el mandato Aplicar cambios registrados por diario (APYJRNCHG) o Eliminar cambios registrados por diario (RMVJRNCHG) para completar o retrotraer las transacciones parciales. x'02' (ROLLBACK_ENDED) - El objeto finalizó una operación de retrotracción con la opción "Finalizar retrotracción" en la pantalla Trabajar con definición de compromiso (WRKCMTDFN). Se recomienda restaurar el objeto, pues no puede utilizarse. Como última opción, puede utilizarse el mandato Cambiar objeto registrado por diario (CHGJRNOBJ) para que pueda utilizarse el objeto. Sin embargo, ello podría dejar al objeto en un estado inconsistente.
QEZLANGID	GRAPHIC (3)	Un ID de tres caracteres que representa el idioma en el que está el nombre del objeto (campo QEZOBJNAM).
QEZLOCAL	SMALLINT	Si un objeto se almacena localmente o se almacena en un sistema remoto. La decisión de si un objeto es local o remoto varía según las reglas del sistema de archivos respectivo. Los objetos de sistemas de archivos que no incorporan un indicador local o remoto, se tratan como remotos. Los valores válidos son: 1 - Los datos del objeto se almacenan localmente. 2 - Los datos del objeto se encuentran en un sistema remoto.
QEZMLTSIG	SMALLINT	Si un objeto tiene más de una firma digital de i5/OS. Los valores válidos son: 0 - El objeto tiene solo una firma digital. 1 - El objeto tiene más de una firma digital. Si el campo QEZSYSSIG tiene el valor 1, como mínimo una de las firmas procede de una fuente en la que el sistema confía.
QEZMODE	INTEGER	La modalidad y el tipo de acceso del archivo. Para obtener más información sobre la modalidad, consulte la API open().

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZMSTGOPT	SMALLINT	Esta opción debería utilizarse para determinar cómo el sistema asigna y utiliza el almacenamiento principal para el objeto especificado. Esta opción solo se puede especificar para archivos continuos en los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario. Los valores válidos son: 0 - El almacenamiento principal se asignará de la forma habitual. Es decir, se asignará y utilizará tanto almacenamiento principal como sea posible. Ello minimiza el número de operaciones de E/S en disco, pues la información se almacena temporalmente en el almacenamiento principal. 1 - El almacenamiento principal se asignará para minimizar el espacio utilizado por el objeto. Es decir, se asignará y utilizará tan poco almacenamiento principal como sea posible. Ello minimiza la utilización de almacenamiento principal al mismo tiempo que aumenta el número de operaciones de E/S en disco pues se almacena menos información temporalmente en el almacenamiento principal. 2 - El sistema determinará dinámicamente la asignación óptima de almacenamiento principal para el objeto según la actividad del sistema y la demanda de almacenamiento principal. Es decir, cuando haya poca demanda de almacenamiento principal, se asignará y utilizará tanto almacenamiento como sea posible para minimizar el número de operaciones de E/S en disco. Y cuando haya mucha demanda de almacenamiento principal, se asignará y utilizará menos almacenamiento principal para minimizar la demanda de almacenamiento principal. Esta opción solo entra en vigor cuando la opción de paginación de la agrupación de almacenamiento es *CALC. Si la opción de paginación de la agrupación de almacenamiento es *FIXED, el comportamiento es el mismo que STG_NORMAL. Esta opción no entra en vigor si se accede al objeto a través de un servidor de archivos. Por el contrario, su comportamiento es el mismo que con STG_NORMAL.
QEZNLNK	INTEGER	El número de enlaces fijos al objeto.
QEZNMCCSID	INTEGER	El CCSID en el que está representado el nombre del objeto (campo QEZOBJNAM).
QEZNONSAV	SMALLINT	Si el objeto puede o no salvarse. Los valores válidos son: 0 - El objeto se salvará. 1 - El objeto no se salvará. Además, si este objeto es un directorio, no se salvará ninguno de los objetos en el subárbol del directorio a menos que se especificaran explícitamente como objeto que se debe salvar. El subárbol incluye todos los subdirectorios y los objetos dentro de dichos subdirectorios.
QEZOBJLEN	INTEGER	Número de bytes que contiene el nombre del objeto (campo QEZOBJNAM).
QEZOBJNAM [1]	VARGRAPHIC (1024)	El nombre del objeto. [2]
QEZOBJTYPE [1]	GRAPHIC(10)	El tipo de objeto.
QEZOFLOW	SMALLINT	Indica si el objeto ha desbordado la agrupación de almacenamiento auxiliar en la que reside. Los valores válidos son: 0 - No se ha desbordado la agrupación de almacenamiento auxiliar. 1 - Se ha desbordado la agrupación de almacenamiento auxiliar.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZOWN [1]	GRAPHIC(10)	El nombre del perfil de usuario que es el propietario del objeto o el siguiente valor especial: *NOUSRPRF - El sistema de archivos de red utiliza este valor especial para indicar que no hay un perfil de usuario en el servidor iSeries local con un ID de usuario (UID) que coincida con el UID del objeto remoto.
QEZOWNPGP	GRAPHIC(10)	El nombre del perfil de usuario que es el grupo primario del objeto o los siguientes valores especiales: *NONE - El objeto no tiene un grupo primario *NOUSRPRF - El sistema de archivos de red utiliza este valor especial para indicar que no hay ningún perfil de usuario en el servidor local con un ID de grupo (GID) que coincida con el GID del objeto remoto.
QEZPCARC	SMALLINT	Si el objeto ha cambiado desde la última vez en que se examinó. 0 - El objeto no ha cambiado. 1 - El objeto ha cambiado.
QEZPCHID [1]	SMALLINT	Si el objeto puede visualizarse utilizando un listado de directorios ordinario. 0 - El objeto no está oculto. 1 - El objeto está oculto.
QEZPCREAD	SMALLINT	Si se puede escribir en el objeto o si este se puede suprimir, si se pueden cambiar o suprimir sus atributos ampliados, o si se puede cambiar su tamaño. Los valores válidos son: 0 - El objeto se puede cambiar. 1 - El objeto no se puede cambiar.
QEZPCSYS	SMALLINT	Si el objeto es un archivo del sistema y se excluye de búsquedas de directorio normales. 0 - El objeto no es un archivo del sistema. 1 - El objeto es un archivo del sistema.
QEZPRMLNK	SMALLINT	Cuando un objeto tiene varios nombres, este campo se establecerá solo para el primer nombre encontrado.
QEZRDEV	BIGINT	Si el objeto representa a un archivo especial de dispositivo, el dispositivo real al que representa.
QEZREGION	GRAPHIC (2)	Un ID de dos caracteres que representa el país del nombre del objeto (campo QEZOBJNAM). Este ID afecta a aquellas acciones que suelen definirse mediante la ubicación de la acción, como el orden de clasificación.
QEZSBINARY	GRAPHIC (1)	Indica si el objeto se exploró en modalidad binaria la última vez. Este campo tendrá uno de los valores siguientes: x'00' - El objeto no se exploró en modalidad binaria. x'01' - El objeto se exploró en modalidad binaria. Si el estado de exploración del objeto es SCAN_SUCCESS, el objeto se exploró satisfactoriamente en modalidad binaria. Si el estado de exploración del objeto es SCAN_FAILURE, el objeto no pudo realizar la exploración en modalidad binaria.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZSCCSID1	INTEGER	Indica si el objeto se exploró en el CCSID listado la última vez que se exploró. Si el estado de exploración del objeto es SCAN_SUCCESS, el objeto se exploró satisfactoriamente en este CCSID. Si el estado de exploración del objeto es SCAN_FAILURE, el objeto finalizó de forma anómala la exploración en este CCSID. Un valor de 0 significa que este campo no es aplicable.
QEZSCCSID2	INTEGER	Indica si el objeto se exploró en el CCSID listado la última vez que se exploró. Si el estado de exploración del objeto es SCAN_SUCCESS, el objeto se exploró satisfactoriamente en este CCSID. Si el estado de exploración del objeto es SCAN_FAILURE, este campo será 0. Un valor de 0 significa que este campo no es aplicable.
QEZSCN	GRAPHIC (1)	Especifica si el objeto se explorará cuando haya programas de salida registrados en cualquiera de los puntos de salida relacionados con la exploración del sistema de archivos integrado. Los valores válidos son: x'00' (SCANNING_NO) - El objeto no se explorará según las reglas descritas en los programas de salida relacionados con la exploración. **Nota:** si no se especifica el valor *NOPOSTRST en Control de exploración de sistemas de archivos (QSCANFSCTL) cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración. x'01' (SCANNING_YES) - El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración si el objeto ha sido modificado o si el software de exploración ha sido actualizado desde la última vez que se exploró el objeto. x'02' (SCANNING_CHGONLY) - El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración solo si el objeto se ha modificado desde la última vez en que se exploró. No se explorará si se ha actualizado el software de exploración. Este atributo solo entra en vigor si para el valor del sistema Control de exploración de sistemas de archivos (QSCANFSCTL) se ha especificado *USEOCOATR. De lo contrario, se considerará que el atributo es SCANNING_YES. **Nota:** si no se especifica el valor *NOPOSTRST en Control de exploración de sistemas de archivos (QSCANFSCTL) cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración.
QEZSIG [1]	SMALLINT	Si un objeto tiene una firma digital de i5/OS. Los valores válidos son: 0 - El objeto no tiene una firma digital de i5/OS. 1 - El objeto tiene una firma digital de i5/OS.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZSSIGDF	GRAPHIC (1)	Las firmas de exploración proporcionan una indicación del nivel de soporte del software de exploración. Si un objeto se encuentra en un grupo de ASP independientes, la firma de exploración del objeto se compara con la firma de exploración del grupo de ASP independientes asociado. Si un objeto no se encuentra en un grupo de ASP independientes, la firma de exploración del objeto se compara con el valor de la firma de exploración global. Este campo tendrá uno de los valores siguientes: x'00' - Las firmas comparadas no son diferentes. x'01' - Las firmas comparadas son diferentes.
QEZSSTATUS	GRAPHIC (1)	El estado de exploración asociado con este objeto. Este campo tendrá uno de los valores siguientes: x'00' (SCAN_REQUIRED) - Es necesaria una exploración del objeto porque todavía no ha sido explorado por los programas de salida relacionados con la exploración o porque se han modificado datos de los objetos o el CCSID desde la última exploración. Algunos ejemplos de modificaciones de datos de los objetos o sel CCSID son: escribir en el objeto, directamente o a través de correlaciones de memoria, truncar el objeto, borrar el objeto y cambiar el atributo CCSID del objeto. x'01' (SCAN_SUCCESS) - El objeto se ha explorado mediante un programa de salida relacionado con la exploración y, en el momento de la última petición de exploración, el objeto no falló la exploración. x'02' (SCAN_FAILURE) - El objeto se ha explorado mediante un programa de salida relacionado con la exploración y, en el momento de la última petición de exploración, el objeto no superó la exploración y no se llevó a cabo la operación. Tras marcar un objeto como anómalo, no volverá a explorarse de nuevo hasta que la firma de exploración del objeto sea distinta a la firma de la clave de exploración global o la firma de la clave de exploración del grupo de ASP independientes, según corresponda. Por lo tanto, las posteriores peticiones de trabajar con el objeto finalizarán de forma anómala con una indicación de anomalía de exploración. Como ejemplos de peticiones que finalizarán de forma anómala pueden citarse: abrir el objeto, cambiar el CCSID del objeto, copiar el objeto. x'05' (SCAN_PENDING_CVN) - El objeto no está en un directorio *TYPE2 y, por lo tanto, no se explorará hasta que se convierta el directorio. x'06' (SCAN_NOT_REQUIRED) - El objeto no requiere ninguna exploración porque se ha identificado para que no sea explorado.
QEZSTGFREE [1]	SMALLINT	Si se han movido los datos del objeto fuera de línea, liberando su almacenamiento en línea. Los valores válidos son: 0 - Los datos del objeto no están fuera de línea. 1 - Los datos del objeto están fuera de línea.

Tabla 8. QAEZDxxxxO (almacenar atributos del objeto) (continuación)

Nombre del campo	Tipo de campo	Descripción del campo
QEZSYSARC	SMALLINT	Si el objeto ha cambiado y es necesario salvarlo. Se activa cuando se actualiza la hora de cambio del objeto y se desactiva cuando se ha salvado el objeto. 0 - El objeto no ha cambiado y no es necesario salvarlo. 1 - El objeto ha cambiado y es necesario salvarlo.
QEZSYSSIG	SMALLINT	Si el objeto fue firmado por una fuente en la que el sistema confía. Los valores válidos son: 0 - Ninguna de las firmas proviene de una fuente en la que el sistema confía. 1 - El objeto ha sido firmado por una fuente en la que el sistema confía. Si el objeto tiene múltiples firmas, como mínimo una de las firmas provenía de una fuente en la que el sistema confía.
QEZUDATE	TIMESTAMP	El número de segundos desde la Época que corresponde a la fecha en que se utilizó el objeto por última vez. Este campo es cero cuando se crea el objeto. Este campo es cero si no se actualizan los datos de utilización para el tipo de i5/OS o el sistema de archivos al que pertenece un objeto.
QEZUDCOUNT	INTEGER	El número de días que se ha utilizado un objeto. La utilización tiene distintos significados según el sistema de archivos específico y según los tipos de objeto individuales a los que se da soporte dentro de un sistema de archivos. La utilización puede indicar la apertura o cierre de un archivo o puede referirse a la adición de enlaces, la redenominación, la restauración o la reserva de un objeto. Este contador se incrementa cada día que se utiliza un objeto y se restaura en cero con la API Qp0lSetAttr().
QEZUDFTYP2	SMALLINT	El formato del archivo por omisión de los archivos continuos (*STMF) creados en el sistema de archivos definido por el usuario. Los valores válidos son: 0 - El formato del archivo continuo es *TYPE1. 1 - El formato del archivo continuo es *TYPE2.
QEZUID	INTEGER	Cada usuario del sistema debe tener un número de identificación de usuario (UID) numérico y exclusivo.
QEZURESET	INTEGER	El número de segundos desde la Época que corresponde a la fecha en que se restauró en cero (0) el contador de días de utilización. Esta fecha se establece en la fecha actual cuando se llama la API Qp0lSetAttr() para que restaure en cero el contador de días de utilización.

Notas:

1. Este campo se incluye en el subconjunto de campos utilizado por el mandato PRTDIRINF.
2. En este campo, solo se almacena el nombre del objeto. El resto del nombre de la vía de acceso se almacena en el campo QEZDIRNAM1 si la longitud del nombre del directorio es inferior a 1 KB (1024 bytes) o en QEZDIRNAM2 si los nombres de directorio son superiores a 1 KB (1024 bytes).

La tabla siguiente es un ejemplo de una tabla que lista los directorios procesados por el mandato RTVDIRINF.

Tabla 9. QAEZDxxxD (almacenar atributos del directorio)

Nombre del campo	Tipo de campo	Descripción del campo
QEZDFID	INTEGER	El ID de archivo del directorio.
QEZDIRFID	GRAPHIC (16)	El ID de archivo del directorio. Un identificador asociado con el objeto. Puede utilizarse el ID de un archivo con Qp0lGetPathFromFileID() para recuperar el nombre de la vía de acceso de un objeto.
QEZDIRFSID	BIGINT	El ID del sistema de archivos del directorio.
QEZDIRGID	BIGINT	El ID de generación.
QEZDIRIDX	INTEGER	Identificador del nombre de la vía de acceso (válido solo para directorios).
QEZDIRLEN [1]	INTEGER	La longitud del nombre de la vía de acceso del directorio.
QEZDIRNAM1 [1]	VARGRAPHIC (1024)	La vía de acceso del directorio padre. Solo se utiliza si la longitud de la vía de acceso es inferior a 1 KB (1024 bytes).
QEZDIRNAM2 [1]	DBCLOB (16M)	La vía de acceso del directorio padre. Solo se utiliza si la longitud de la vía de acceso es superior a 1 KB (1024 bytes) de longitud. Puede almacenar vías de acceso de hasta 16 MB de longitud.
QEZDRCCSID	INTEGER	El CCSID del directorio.
QEZDREGION	GRAPHIC (2)	El ID de zona de la vía de acceso del directorio.
QEZLANGID	GRAPHIC (3)	El ID de idioma de la vía de acceso del directorio.
Nota: • Este campo se incluye en el subconjunto de campos utilizado por el mandato PRTDIRINF.		

La tabla siguiente muestra la información que almacena el mandato RTVDIRINF sobre los archivos que ha creado al ejecutarse. Si el archivo que contiene esta información no existe, el mandato RTVDIRINF lo crea; cuando se ejecuta el mandato en varias ocasiones, la información se anexa al archivo existente. El mandato PRTDIRINF utiliza esta información para determinar qué archivos de base de datos se utilizaron para almacenar la información recuperada por instancias diferentes del mandato RTVDIRINF.

Tabla 10. QUSRSYS/QAEZDBFILE (almacenar archivos creados)

Nombre del campo	Tipo de campo	Descripción del campo
QEZDIRFILE [1]	VARGRAPHIC (20)	El nombre del archivo generado para almacenar los índices del directorio.
QEZDIRSRC	VARGRAPHIC (5000)	Vía de acceso especificada en el parámetro DIR (RTVDIRINF).
QEZENDTIME	TIMESTAMP	La fecha/hora en la que se completó RTVDIRINF.
QEZLIB [1]	VARGRAPHIC (20)	La biblioteca donde residen los archivos generados.
QEZOBJFILE [1]	VARGRAPHIC (20)	El nombre del archivo generado para almacenar los atributos del objeto.
QEZPLANGID	GRAPHIC (3)	El ID de idioma de la vía de acceso.
QEZPRCCSID	INTEGER	El CCSID de la vía de acceso.
QEZPREGION	GRAPHIC (2)	El ID local de la vía de acceso.
QEZSTRTIME	TIMESTAMP	La fecha/hora en la que se sometió RTVDIRINF.
Nota: • Este campo se incluye en el subconjunto de campos utilizado por el mandato PRTDIRINF.		

Información relacionada

Mandato Recuperar información de directorios (RTVDIRINF)

API Qp01GetPathFromID()

API Qp0lSetAttr()

Mandato Aplicar cambios registrados por diario (APYJRNCHG)

Mandato Eliminar cambios registrados por diario (RMVJRNCHG)

Mandato Cambiar objeto registrado por diario (CHGJRNOBJ)

Mandato Imprimir información de directorio (PRTDIRINF)

Acceso a los datos de RTVDIRINF:

Existen varias opciones para acceder a los datos de las tablas.

Más abajo aparecen las formas en las que se puede acceder a los datos creados por el mandato Recuperar información de directorio (RTVDIRINF):

- Utilización del mandato Imprimir información de directorio (PRTDIRINF)

  Este mandato se utiliza para imprimir información de directorio sobre los objetos e información de directorio en el sistema de archivos integrado. La información que se imprimirá ya se encuentra almacenada en el archivo de base de datos especificado por el usuario en el mandato RTVDIRINF.
- Utilización de cualquier programa o mandato facilitado por IBM que pueda ejecutar consultas a través de una tabla de DB2 en iSeries.

  Algunas de las herramientas más habituales son el mandato Iniciar sesión interactiva SQL (STRSQL) e iSeries Navigator.

  Por ejemplo, si desea seleccionar objetos en una vía de acceso específica (anteriormente recogida por el mandato RTVDIRINF) que tenga un tamaño de asignación superior a 10kb, puede ejecutar una consulta de este modo:

  SELECT QEZOBJNAM, QEZALCSIZE FROM library_name/QAEZDxxxxO WHERE QEZALCSIZE > 10240
- Puede crear sus propios programas y acceder a las tablas de base de datos utilizando cualquier método de BD válido.

  Información relacionada

  Mandato Imprimir información de directorio (PRTDIRINF)

  Mandato Iniciar SQL (STRSQL)

  SQL incorporada

  CLI de SQL

Utilización de los datos de RTVDIRINF:

A continuación se incluyen ejemplos que muestran porqué son importantes los datos o cómo puede utilizar los datos resultantes de cada una de las tres tablas.

- Para Tabla 8 en la página 80, pueden efectuarse consultas para crear informes o estadísticas basados en cualquiera de los campos de esta tabla. PRTDIRINF no incluye informes basados en todos los campos. Por el contrario, se utilizará un subconjunto.
- Los datos de Tabla 9 en la página 91 contienen todos los directorios de la vía de acceso especificada en el parámetro DIR del mandato RTVDIRINF. Si desea conocer atributos específicos sobre el nombre de la vía de acceso, por ejemplo el CCSID, el ID de idioma o la longitud, entonces estos datos serían útiles. Además, cada directorio almacenado en esta tabla tiene un valor exclusivo o un índice que lo identifica. En Tabla 8 en la página 80, puede encontrar el mismo campo, QEZDIRIDX, que le informará sobre qué objetos pertenecen a qué directorio. Para averiguar qué objetos pertenecen a qué directorio, puede realizar una consulta utilizando uniones. Por ejemplo, la siguiente declaración de consulta selecciona los nombres de todos los objetos existentes en el directorio ″/MYDIR″:

  ```
  SELECT QEZOBJNAM FROM library_name/QAEZxxxxO, library_name/QAEZxxxxD WHERE QEZDIRNAM1 = "/MYDIR" AND
  library_name/QAEZxxxxO.QEZDIRIDX=library_name/QAEZxxxxD.QEZDIRIDX
  ```

- Tabla 10 en la página 91 se utiliza principalmente por el mandato PRTDIRINF para obtener información específica sobre ejecuciones de RTVDIRINF. Algunos ejemplos son: los nombres de las tablas creadas, la biblioteca en la que residen las tablas, y la hora de inicio y finalización del proceso. Podría utilizar esta tabla para saber cuándo se emitió un RTVDIRINF o qué tablas deben examinarse para las consultas.

Acceso utilizando API

Puede utilizar interfaces de programa de aplicación (API) para acceder al sistema de archivos integrado.

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Acceso utilizando iSeries Navigator

iSeries Navigator es la interfaz gráfica de usuario que permite gestionar y administrar sus sistemas desde su escritorio Windows. 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 136
- "Eliminar una carpeta" en la página 136
- "Reincorporar un archivo" en la página 135
- "Reservar un archivo" en la página 135
- "Establecer permisos" en la página 138
- "Configurar la conversión de archivos de texto" en la página 138
- "Enviar un archivo o carpeta a otro sistema" en la página 138
- "Cambiar opciones de una definición del paquete" en la página 139
- "Planificar fecha y hora en que enviar el archivo o carpeta" en la página 139
- "Establecer si debe o no explorarse los objetos" en la página 141

Trabajar con compartimientos de archivos

- "Crear un compartimiento de archivos" en la página 140
- "Cambiar un compartimiento de archivos" en la página 140

Trabajar con sistemas de archivos definidos por el usuario

- "Crear un nuevo sistema de archivos definido por el usuario" en la página 140
- "Montar un sistema de archivos definido por el usuario" en la página 140
- "Desmontar un sistema de archivos definido por el usuario" en la página 141

Registro por diario de objetos

- "Iniciar registro por diario" en la página 107
- "Finalizar registro por diario" en la página 108

Referencia relacionada

"Acceso mediante un PC" en la página 96
Si el PC está conectado a un servidor iSeries, se podrá interaccionar con los directorios y objetos del sistema de archivos integrado como si estuviesen almacenados en el PC.

Acceso utilizando iSeries NetServer

El soporte de iSeries para el Entorno de Red de Windows (iSeries NetServer) es una función de i5/OS que permite a los clientes Windows el acceso a las vías de acceso del directorio compartido y las colas de salida compartidas de i5/OS. iSeries NetServer permite a los PC que ejecutan software de Windows el acceso sin problemas a datos e impresoras gestionados por 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 iSeries NetServer.

Los clientes Linux con el software de cliente Samba instalado también pueden acceder sin problemas a datos e impresoras a través de iSeries NetServer. Los directorios de iSeries NetServer compartidos se pueden montar en clientes Linux como sistemas de archivos Samba (smbfs) de un modo similar al utilizado para montar sistemas de archivos NFS que se han exportado desde iSeries.

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

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

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 de iSeries NetServer. Por ejemplo, puede introducir la siguiente sintaxis: `\\QSYSTEM1\Nombre_compartido`

   Nota: `QSYSTEM1` es el nombre de sistema de iSeries NetServer, y `Nombre_compartido` es el nombre del compartimiento de archivos que desea utilizar.
5. Pulse **Aceptar**.

Nota: Si se conecta mediante iSeries NetServer, el nombre del servidor puede ser distinto del nombre utilizado por iSeries Access Family. Pro ejemplo, el nombre de iSeries NetServer 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 Family 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 iSeries NetServer. 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 `fredsdir`, un usuario podrá acceder a dicho directorio desde el PC con el nombre `\\QAS400X\FREDSDIR`, o desde un cliente LINUX con el nombre `//qas400x/fredsdir`.

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″ (/). Para obtener más información, consulte el apartado “Trasladar archivos o carpetas a otro sistema de archivos” en la página 136.

Referencia relacionada

“Acceso mediante un PC” en la página 96
Si el PC está conectado a un servidor iSeries, se podrá interaccionar con los directorios y objetos del sistema de archivos integrado como si estuviesen almacenados en el PC.

Información relacionada

iSeries NetServer

Compartimientos de archivos de iSeries NetServer

Acceso al compartimiento de archivos de iSeries NetServer con un cliente de PC Windows

Acceso utilizando el protocolo de transferencia de archivos

El cliente de FTP (protocolo de transferencia de archivos) permite transferir los archivos que se encuentran en el servidor iSeries, incluidos los de los sistemas de archivos ″raíz″ (/), QOpenSys, QSYS.LIB, QSYS.LIB de ASP independiente, QOPT y 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 una modalidad por lotes desatendida en la que los submandatos de cliente se leen en un archivo y las respuestas a estos submandatos se escriben en un archivo. También incluye otras funciones para manipular 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 NetWare (QNetWare)
- Sistema de archivos iSeries NetClient (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 objetos, como los programas (*PGM). No obstante, se pueden salvar otros tipos de objetos en un archivo de salvar, transferir el archivo de salvar y luego restaurar los objetos.

Referencia relacionada

“Acceso mediante un PC” en la página 96
Si el PC está conectado a un servidor iSeries, se podrá interaccionar con los directorios y objetos del sistema de archivos integrado como si estuviesen almacenados en el PC.

Información relacionada

FTP

Transferencia de archivos con FTP

Acceso mediante un PC

Si el PC está conectado a un servidor iSeries, se podrá interaccionar con los directorios y objetos del sistema de archivos integrado como si estuviesen almacenados en el PC.

Puede copiar objetos de un directorio a otro utilizando la opción de arrastras 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 servidor iSeries 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 Family 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.

Según el tipo de objeto, pueden utilizarse las interfaces del PC y 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 servidor iSeries 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 iSeries Navigator (una parte de iSeries Access Family). El PC utiliza clientes de compartimiento de archivos de Windows para el acceso a iSeries NetServer, que se ejecuta en el servidor iSeries.

Conceptos relacionados

"Acceso utilizando iSeries Navigator" en la página 93
iSeries Navigator es la interfaz gráfica de usuario que permite gestionar y administrar sus sistemas desde su escritorio Windows. iSeries Navigator facilita el manejo y la administración del sistema y mejora la productividad.

Tareas relacionadas

"Acceso utilizando iSeries NetServer" en la página 94
El soporte de iSeries para el Entorno de Red de Windows (iSeries NetServer) es una función de i5/OS que permite a los clientes Windows el acceso a las vías de acceso del directorio compartido y las colas de salida compartidas de i5/OS. iSeries NetServer permite a los PC que ejecutan software de Windows el acceso sin problemas a datos e impresoras gestionados por iSeries.

Referencia relacionada

"Acceso utilizando el protocolo de transferencia de archivos" en la página 95
El cliente de FTP (protocolo de transferencia de archivos) permite transferir los archivos que se encuentran en el servidor iSeries, incluidos los de los sistemas de archivos "raíz" (/), QOpenSys, QSYS.LIB, QSYS.LIB de ASP independiente, QOPT y QFileSvr.400.

Conversión de directorios de *TYPE1 a *TYPE2

Los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

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.

Poco después de la instalación de i5/OS V5R3M0 o un release posterior, el sistema empieza a convertir automáticamente a directorios *TYPE2 cualquier sistema de archivos que todavía no de soporte a los directorios *TYPE2. Esta conversión no debería afectar demasiado a la actividad del sistema.

Conceptos relacionados

“Directorios *TYPE2” en la página 10
Los sistemas de archivos ″raíz″ (/), QOpenSys y 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.

Visión general de la conversión de *TYPE1 a *TYPE2

Los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

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. Además de mejorar el rendimiento y la fiabilidad, existen funciones nuevas, como el soporte de la exploración del sistema de archivos integrado, que solo están disponibles para los objetos de los directorios *TYPE2. Para obtener más información, consulte el apartado “Soporte para la exploración” en la página 21.

Poco después de la instalación de un sistema operativo i5/OS V5R3M0 o un release posterior, el sistema empezará a convertir automáticamente a directorios *TYPE2 cualquier sistema de archivos que todavía no de soporte a los directorios *TYPE2. Esta conversión no debería afectar demasiado a la actividad del sistema, pues se ejecutará como trabajo de fondo de prioridad baja.

Si todavía no se ha completado la función de conversión, y el sistema tiene una IPL normal o anormal, la función de conversión finalizará cuando se complete la IPL. La conversión se reiniciará en cada IPL hasta que se hayan convertido por completo todos los sistemas de archivos apropiados.

Los sistemas de archivos apropiados para esta conversión automática son los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario para las ASP del 1 al 32.

Nota: Puede impedir la conversión automática a directorios *TYPE2 si convierte los sistemas de archivos antes de instalar el sistema operativo V5R3M0 o un release posterior.

Conceptos relacionados

“Directorios *TYPE2” en la página 10
Los sistemas de archivos ″raíz″ (/), QOpenSys y 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.

Referencia relacionada

“Determinar estado de conversión” en la página 98
Poco después de la instalación de un sistema operativo i5/OS V5R3M0 o un release posterior, el sistema empezará a convertir automáticamente a directorios *TYPE2 cualquier sistema de archivos que todavía no de soporte a los directorios *TYPE2. Este proceso de conversión tendrá lugar en una hebra secundaria del trabajo del sistema QFILESYS1.

“Consejos: ASP independiente” en la página 101
Los sistemas de archivos definidos por usuario de una ASP, si todavía no se han convertido al formato de directorios *TYPE2, se convertirán la primera vez que se active la ASP independiente en un sistema instalado con la versión V5R2 o posterior del sistema operativo.

Consideraciones sobre la conversión

A continuación, se describen las consideraciones que se deben tener en cuenta durante el proceso de conversión.

Determinar estado de conversión

Poco después de la instalación de un sistema operativo i5/OS V5R3M0 o un release posterior, el sistema empezará a convertir automáticamente a directorios *TYPE2 cualquier sistema de archivos que todavía no de soporte a los directorios *TYPE2. Este proceso de conversión tendrá lugar en una hebra secundaria del trabajo del sistema QFILESYS1.

Para determinar el estado del proceso de conversión, puede utilizar el mandato Convertir directorio (CVTDIR) del siguiente modo:

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

Esta invocación del mandato CVTDIR muestra el formato actual de directorios para los sistemas de archivos "raíz" (/), QOpenSys y UDFS, e indica si actualmente se está convirtiendo el sistema de archivos. Además, muestra la prioridad actual de la función de conversión, el sistema de archivos que actualmente convierte el sistema, el número de enlaces que se han procesado para dicho sistema de archivos, y el porcentaje de directorios que se han procesado para ese sistema de archivos. El sistema inicia la función de conversión con una prioridad muy baja (99), por lo tanto la función de conversión no afecta demasiado a la actividad del sistema. Sin embargo, puede cambiarse la prioridad de la función de conversión utilizando el valor *CHGPTY para el parámetro OPTION del mandato CVTDIR. Consulte CVTDIR para obtener más información sobre esta especificación de parámetros.

Puesto que el trabajo QFILESYS1 procesa la conversión, es posible ver las anotaciones de trabajo QFILESYS1 para los mensajes que advierten de cualquier problema durante la conversión. Además, se envían varios mensajes de progreso sobre las conversiones del sistema de archivos. Estos mensajes incluyen información como: el sistema de archivos que se está convirtiendo, el número de enlaces que se han procesado en el sistema de archivos, el porcentaje de directorios que se han procesado en el sistema de archivos, etc. Todos los mensajes de error y muchos mensajes de progreso se envían también a la cola de mensajes QSYSOPR. Por lo tanto, para poder consultarlos en el futuro, sería adecuado asegurarse que se preservan las anotaciones QHST o las anotaciones de trabajo QFILESYS1, que incluirán estos mensajes. Tras haber convertido totalmente los sistemas de archivos y cuando el sistema de archivos integrado esté funcionando del modo esperado, podrá borrar esta información histórica.

Conceptos relacionados

"Visión general de la conversión de *TYPE1 a *TYPE2" en la página 97
Los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

Información relacionada

Mandato Convertir directorio (CVTDIR)

Crear perfiles de usuario

La función de conversión crea un perfil de usuario que se utiliza durante la ejecución de la función de conversión. Este perfil de usuario tiene el nombre QP0FCWA. La función de conversión lo utiliza para tomar la propiedad de directorios convertidos del sistema de archivos cuando el propietario original no es capaz de tomar la propiedad de sus directorios.

Si es posible, el perfil de usuario se suprime al finalizar la conversión. El mensaje CPIA08B se envía a las anotaciones de trabajo QFILESYS1 y a la cola de mensajes QSYSOPR si se otorga la propiedad de un directorio a este perfil de usuario.

Información relacionada

"Cambiar el propietario de un directorio" en la página 100
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 el perfil de usuario alternativo.

Redenominar objetos

Los directorios *TYPE2 requieren que los nombres de enlace sean nombres UTF-16 válidos.

La regla de denominación de los directorios * TYPE2 es distinta de la de 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 no válido o duplicado, el nombre se convierte en un nombre UTF-16 válido y exclusivo, y se envía el mensaje CPIA08A a las anotaciones de trabajo QFILESYS1 y a la cola de mensajes QSYSOPR con el nombre original y el nombre nuevo. Los caracteres combinados o las parejas de caracteres sustitutos no válidas de un nombre pueden provocar el cambio del nombre de un objeto.

Para obtener más información sobre UTF-16, consulte la página de presentación de Unicode www.unicode.org).

Caracteres combinados:

Algunos caracteres pueden estar formados por varios caracteres Unicode.

Por ejemplo, los caracteres que tienen acento (por ejemplo, é o à) o una diéresis (por ejemplo, ä o ö) deben cambiarse, o *normalizarse*, a un formato común antes de almacenarse en el directorio de forma que todos los objetos tengan un nombre exclusivo. 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 (solo un carácter parcial), es un nombre no válido. 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 no válidos, se cambiará el nombre antes de enlazar el objeto con 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.

Cambiar el almacenamiento máximo para un perfil de usuario:

Durante el proceso de conversión de directorios, varios directorios que momentáneamente existen en ambos formatos al mismo tiempo pertenecen al mismo perfil de usuario.

Si durante el proceso de conversión se alcanza el límite máximo de almacenamiento para el perfil de usuario, este límite se incrementa. El mensaje CPIA08C se envía a las anotaciones de trabajo QFILESYS1 y a la cola de mensajes QSYSOPR.

Cambiar 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 el perfil de usuario alternativo.

El mensaje CPIA08B se envía a las anotaciones de trabajo QFILESYS1 y a la cola de mensajes QSYSOPR y la conversión continúa.

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 el perfil de usuario alternativo. El mensaje CPIA08B se envía a las anotaciones de trabajo QFILESYS1 y a la cola de mensajes QSYSOPR y la conversión continúa.

Referencia relacionada

"Crear perfiles de usuario" en la página 98
La función de conversión crea un perfil de usuario que se utiliza durante la ejecución de la función de conversión. Este perfil de usuario tiene el nombre QP0FCWA. La función de conversión lo utiliza para tomar la propiedad de directorios convertidos del sistema de archivos cuando el propietario original no es capaz de tomar la propiedad de sus directorios.

Requisitos de almacenamiento auxiliar

Deben considerarse los requisitos de almacenamiento auxiliar cuando se convierten los directorios de un sistema de archivos al formato *TYPE2.

Existen varias consideraciones a tener en cuenta sobre 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 que los directorios posean simultáneamente la versión *TYPE1 y la versión *TYPE2.

Nota: Antes de instalar el sistema operativo i5/OS V5R3M0 o un release posterior, puede ejecutar la opción *ESTIMATE en el mandato de OS/400 V5R2 (CVTDIR), ya que puede proporcionar una estimación conservadora de la cantidad de almacenamiento auxiliar que se necesita durante la conversión.

Información relacionada

Mandato Convertir directorio (CVTDIR)

Consejos: enlace simbólico

Los enlaces simbólicos son objetos del sistema de archivos integrado que contienen una vía de acceso a otro objeto.

Se dan algunas situaciones durante la conversión en las que se puede cambiar 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.

Conceptos relacionados

"Enlace" en la página 12
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.

Referencia relacionada

"Redenominar objetos" en la página 98
Los directorios *TYPE2 requieren que los nombres de enlace sean nombres UTF-16 válidos.

Información relacionada

Crear enlace simbólico (symlink())

Consejos: ASP independiente

Los sistemas de archivos definidos por usuario de una ASP, si todavía no se han convertido al formato de directorios *TYPE2, se convertirán la primera vez que se active la ASP independiente en un sistema instalado con la versión V5R2 o posterior del sistema operativo.

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 sistema operativo que ejecuta V5R2 o posterior, ejecute la siguiente API en el sistema operativo V5R1 al activar la ASP independiente (con el nombre `ASP_NAME`):

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

Conceptos relacionados

"Visión general de la conversión de *TYPE1 a *TYPE2" en la página 97
Los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario (UDFS) del sistema de archivos integrado admiten el formato de los directorios *TYPE2 a partir de OS/400 V5R1.

Consejos: salvar y restaurar

Los directorios que existen como *TYPE1 pueden salvarse y restaurarse en un sistema de archivos que se ha convertido a *TYPE2.

Del mismo modo, es posible salvar 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.

Consejos: reclamar objetos de sistema de archivos integrado

Mientras el sistema está convirtiendo los sistemas de archivos ″raíz″ (/), QOpenSys y los sistemas de archivos UDFS de la ASP de usuario para dar soporte al formato de directorios *TYPE2, los mandatos Reclamar almacenamiento (RCLSTG) y Reclamar enlaces de objetos (RCLLNK) no pueden ejecutarse en ningún directorio del sistema de archivos integrado, inclusive los de las ASP independientes.

Puede utilizarse el valor del parámetro OMIT(*DIR) en el mandato RCLSTG para omitir directorios del sistema de archivos integrado y permitir reclamar los objetos que no están relacionados con el sistema de archivos integrado.

Conceptos relacionados

"Reclamar los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario" en la página 109
Para reclamar los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario, puede utilizar los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG).

Información relacionada

Mandato Reclamar almacenamiento (RCLSTG)

Mandato Reclamar enlaces de objetos (RCLLNK)

Exploración del sistema de archivos integrado

Los objetos de los sistemas de archivos ″raíz″ (/), QOpenSys y UDFS de la ASP de usuario no se explorarán utilizando los puntos de salida relacionados con la exploración del sistema de archivos integrado hasta que los sistemas de archivos se hayan convertido por completo al formato de directorios *TYPE2.

Pueden establecerse los atributos relacionados con la exploración para los objetos de los directorios *TYPE1 y *TYPE2 para especificar si los objetos deben explorarse o no, incluso si el sistema de archivos no se ha convertido por completo.

Cuando el sistema convierte objetos del formato de directorios *TYPE1 al formato de directorios *TYPE2, se tiene en cuenta el valor del sistema de control de la exploración `Explorar en el siguiente acceso tras la restauración del objeto` como si el objeto convertido hubiera sido restaurado. Por ejemplo, si se especifica el valor `Explorar en el siguiente acceso tras la restauración del objeto` mientras la conversión está en curso, un objeto que estuviera en un directorio *TYPE1, y que tuviera especificado el atributo `el objeto no se explorará`, se explorará como mínimo una vez tras la conversión total del sistema de archivos.

Conceptos relacionados

“Soporte para la exploración” en la página 21
Con iSeries, puede explorar objetos del sistema de archivos integrado.

“Valores del sistema relacionados” en la página 23
Existen dos valores del sistema relacionados con este soporte de exploración. Puede utilizar estos dos valores del sistema para definir el entorno de exploración que desea para su servidor.

Registro por diario de objetos

La finalidad primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se salvó por última vez. Además, un uso importante del registro por diario es duplicar en otro sistema los cambios realizados en un objeto, para mejorar la disponibilidad o equilibrar las cargas de trabajo.

En este apartado se ofrece 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.

Información relacionada

Gestión por diario

Visión general del registro por diario

En estos temas se hace una introducción al soporte del registro por diario de objetos del sistema de archivos integrado.

Información relacionada

Gestión por diario

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 salvó 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 duplicación en otro sistema de los cambios realizados en un objeto, para mejorar la disponibilidad o equilibrar las cargas de trabajo.
- 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 sistema de archivos integrado se pueden registrar por diario archivos continuos, directorios y enlaces simbólicos. Solo se admiten los objetos de los sistemas de archivos ″raíz″ (/), QOpenSys y UDFS.

Conceptos relacionados

"Objetos que debería registrar por diario"
Debe tener en cuenta varias cuestiones cuando decida si va a registrar por diario un objeto del sistema de archivos integrado.

Objetos que debería registrar por diario

Debe tener en cuenta varias cuestiones cuando decida si va a registrar por diario un objeto del sistema de archivos integrado.

Tenga en cuenta las siguientes preguntas para determinar qué objetos debe registrar por diario:

- ¿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 escribir 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.

Conceptos relacionados

"Gestión por diario" en la página 102
La principal finalidad de la gestión por diario es permitir la recuperación de los cambios realizados en un objeto desde que se salvó por última vez.

Objetos de 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 i5/OS.

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 Cl o API) o utilizando iSeries Navigator. Se puede Iniciar el registro por diario y Finalizar el registro por diario mediante iSeries Navigator, así como mostrar información del estado del registro por diario.

| **Nota:** los archivos continuos correlacionados con la memoria, los archivos de volumen virtual y los archivos continuos que se utilizan como espacios de almacenamiento de red IXS (Integrated xSeries Server para iSeries) no se pueden registrar por diario. No es posible registrar por diario los directorios que pueden contener objetos de archivo especial de bloqueo. Por ejemplo: /dev/QASP01, /dev/QASP22 y /dev/IASPNAME.

En la siguiente lista se resume el soporte de registro por diario en el 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, incluidos los de Iniciar registro por diario, Finalizar registro por diario, Cambiar registro por diario y Aplicar cambios registrados por diario, se pueden ejecutar 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, redenominar 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 redenominar o trasladar el objeto) harán que el sistema inicie automáticamente el registro por diario.

  Nota: Consideraciones sobre el atributo Heredar registro por diario:
  - Si se redenomina un 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.
  - Cuando un directorio se desplaza a un directorio para el que se ha activado el atributo de heredar registro por diario, solo se inicia el registro por diario para el directorio que se ha desplazado, si corresponde. Los objetos del interior de dicho directorio desplazado no se ven afectados.
  - Si se restaura un objeto a un directorio que tiene activado el atributo de heredar registro por diario, no se inicia el registro por diario para dicho objeto si el objeto se ha registrado por diario alguna vez.
  - Cuando se utiliza el mandato Aplicar cambios registrados por diario (APYJRNCHG), no se utiliza el valor actual del atributo Heredar registro por diario para ningún directorio. Por el contrario, se iniciará o no el registro por diario para cualquier objeto que se cree como parte de la aplicación basándose en qué sucedió durante la actividad de tiempo de ejecución que se está aplicando.
- 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 escritura soportado por las distintas interfaces de escritura es 2 GB - 1. El tamaño máximo de entrada de diario si se especifica RCVSIZOPT (*MAXOPT2 o *MAXOPT3) es 4.000.000.000 de bytes. De lo contrario, el tamaño máximo de entrada de diario es 15.761.440 de bytes. Si registra por diario el archivo continuo y tiene operaciones de escritura que sobrepasan el límite de los 15.761.440 de bytes, deberá utilizar el soporte de *MAXOPT2 o de *MAXOPT3 para evitar que se produzcan errores.

Para obtener más información acerca del diseño de las diferentes entradas de diario, el archivo include de C, qp0ljrnl.h, que se encuentra en el miembro QSYSINC/H (QP0LJRNL), contiene detalles sobre el contenido de los datos de las entradas de diario del sistema de archivos integrado y de sus formatos.

Conceptos relacionados

"Archivo continuo" en la página 18
Un *archivo continuo* es una secuencia de bytes accesible aleatoriamente, sin ninguna otra estructura impuesta por el sistema.

"Directorio" en la página 4
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.

"Enlace simbólico" en la página 15
Un *enlace simbólico*, también denominado enlace dinámico, es un nombre de vía de acceso contenido en un archivo.

Tareas relacionadas

"Iniciar registro por diario" en la página 107
Siga estos pasos para iniciar el registro por diario en un objeto a través del iSeries Navigator.

"Finalizar registro por diario" en la página 108
Después de iniciar el registro por diario en un objeto, si por algún motivo desea finalizar el registro por diario en dicho objeto, puede utilizar los pasos que se describen en este tema.

"Cambiar registro por diario" en la página 108
Después de iniciar el registro por diario en un objeto, si por algún motivo desea cambiar los atributos de diario del objeto sin finalizar y reiniciar el registro por diario, puede utilizar el mandato Cambiar objeto registrado por diario (CHGJRNOBJ), que permite cambiar los objetos registrados por diario.

Información relacionada

Gestión por diario

Buscador de información de entrada de diario

Operaciones registradas en diario

Estas operaciones solo 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:

- Escribir o borrar datos
- Truncar y ampliar un archivo
- Datos de archivo forzados
- Salvar 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
- Mandato Cambiar objeto registrado por diario (CHGJRNOBJ)
- Finalizar registro por diario
- Iniciar el mandato Aplicar cambios registrados por diario (APYJRNCHG)

- Finalizar el mandato Aplicar cambios registrados por diario (APYJRNCHG)
- Salvar
- Restaurar

Información relacionada

Gestión por diario

Buscador de información de entrada de diario

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 retrotracción (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 — Redenominar/Mover
- B4 — Desenlazar (directorio padre)
- B5 — Desenlazar (enlace)
- B7 — Información de autorizaciones sobre objetos creados
- FA — Cambiar atributo
- JT — Iniciar registro por diario (solo 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 o reproducen la operación, generalmente solo 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 solo se dará en operaciones que den como resultado una entrada de diario de retrotracción (C — RB).

Este caso solo 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.

Información relacionada

Buscador de información de entrada de diario

Consideraciones para múltiples enlaces fijos y el registro por diario

Si posee múltiples enlaces fijos a un objeto del sistema de archivos integrado registrado por diario, todos los enlaces deben salvarse y restaurarse conjuntamente de forma que se mantenga el enlace y la información asociada del diario.

Si se especifican nombres en algunos de los mandatos relacionados con el diario y si los nombres son en realidad múltiples enlaces fijos, en ese caso el objeto solo se ejecutará "una vez". El resto de enlaces fijos se ignoran.

Puesto que múltiples enlaces fijos se refieren al mismo objeto y la entrada de diario solo tiene el identificador de archivo (ID de archivo), que es el mismo para el objeto, cualquier interfaz de diario que muestre el nombre de la vía de acceso, por ejemplo, Visualizar diario (DSPJRN), solo mostrará un nombre de enlace para el objeto. Sin embargo, no debería ocasionar ningún problema pues es posible trabajar con un objeto mediante cualquier nombre y obtener el mismo resultado.

Conceptos relacionados

"Enlace fijo" en la página 14
Un *enlace fijo*, en ocasiones denominado simplemente enlace, no puede existir a menos que esté enlazado con un objeto real.

Iniciar registro por diario

Siga estos pasos para iniciar el registro por diario en un objeto a través del iSeries Navigator.

1. Despliegue el sistema en **iSeries Navigator**.
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 iniciar el registro por diario en un objeto a través de la interfaz basada en caracteres, puede utilizarse el mandato Iniciar registro por diario (STRJRN) o la API QjoStartJournal.

Conceptos relacionados

"Objetos de sistema de archivos integrado registrados por diario" en la página 103
Algunos tipos de objetos del sistema de archivos integrado pueden registrarse por diario utilizando el soporte del registro por diario de i5/OS.

Tareas relacionadas

"Cambiar registro por diario" en la página 108
Después de iniciar el registro por diario en un objeto, si por algún motivo desea cambiar los atributos de diario del objeto sin finalizar y reiniciar el registro por diario, puede utilizar el mandato Cambiar objeto registrado por diario (CHGJRNOBJ), que permite cambiar los objetos registrados por diario.

"Finalizar registro por diario" en la página 108
Después de iniciar el registro por diario en un objeto, si por algún motivo desea finalizar el registro por diario en dicho objeto, puede utilizar los pasos que se describen en este tema.

Información relacionada

Mandato Iniciar registro por diario (STRJRN)

API QjoStartJournal

Gestión por diario

Cambiar registro por diario

Después de iniciar el registro por diario en un objeto, si por algún motivo desea cambiar los atributos de diario del objeto sin finalizar y reiniciar el registro por diario, puede utilizar el mandato Cambiar objeto registrado por diario (CHGJRNOBJ), que permite cambiar los objetos registrados por diario.

Conceptos relacionados

"Objetos de sistema de archivos integrado registrados por diario" en la página 103
Algunos tipos de objetos del sistema de archivos integrado pueden registrarse por diario utilizando el soporte del registro por diario de i5/OS.

Tareas relacionadas

"Iniciar registro por diario" en la página 107
Siga estos pasos para iniciar el registro por diario en un objeto a través del iSeries Navigator.

"Finalizar registro por diario"
Después de iniciar el registro por diario en un objeto, si por algún motivo desea finalizar el registro por diario en dicho objeto, puede utilizar los pasos que se describen en este tema.

Información relacionada

Mandato Cambiar objeto registrado por diario (CHGJRNOBJ)

Finalizar registro por diario

Después de iniciar el registro por diario en un objeto, si por algún motivo desea finalizar el registro por diario en dicho objeto, puede utilizar los pasos que se describen en este tema.

Para finalizar el registro por diario en un objeto a través de iSeries Navigator, siga estos pasos:

1. Despliegue 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 iniciar el registro por diario en un objeto a través de la interfaz basada en caracteres, puede utilizarse el mandato Finalizar registro por diario (ENDJRN) o la API QjoEndJournal.

Conceptos relacionados

"Objetos de sistema de archivos integrado registrados por diario" en la página 103
Algunos tipos de objetos del sistema de archivos integrado pueden registrarse por diario utilizando el soporte del registro por diario de i5/OS.

Tareas relacionadas

"Cambiar registro por diario"
Después de iniciar el registro por diario en un objeto, si por algún motivo desea cambiar los atributos de diario del objeto sin finalizar y reiniciar el registro por diario, puede utilizar el mandato Cambiar objeto registrado por diario (CHGJRNOBJ), que permite cambiar los objetos registrados por diario.

"Iniciar registro por diario" en la página 107
Siga estos pasos para iniciar el registro por diario en un objeto a través del iSeries Navigator.

Información relacionada

Mandato Finalizar registro por diario (ENDJRN)

API QjoEndJournal

Gestión por diario

Reclamar los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario

Para reclamar los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por el usuario, puede utilizar los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG).

Con los mandatos RCLLNK y RCLSTG puede realizar las siguientes tareas:

- Corregir problemas de perfiles de usuario de objetos
- Corregir problemas del sistema de archivos definido por el usuario
- Corregir problemas de objetos internos
- Eliminar enlaces de objetos no válidos
- Manejar objetos dañados
- Crear objetos del sistema que faltan
- Corregir problemas de sistemas de archivos internos (solo para RCLSTG)
- Encontrar objetos perdidos (solo para RCLSTG)

Referencia relacionada

“Consejos: reclamar objetos de sistema de archivos integrado” en la página 101
Mientras el sistema está convirtiendo los sistemas de archivos ″raíz″ (/), QOpenSys y los sistemas de archivos UDFS de la ASP de usuario para dar soporte al formato de directorios *TYPE2, los mandatos Reclamar almacenamiento (RCLSTG) y Reclamar enlaces de objetos (RCLLNK) no pueden ejecutarse en ningún directorio del sistema de archivos integrado, inclusive los de las ASP independientes.

Comparación de los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG)

Puede utilizar los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG) para corregir problemas en los sistemas de archivos “raíz” (/), QOpenSys y definidos por el usuario.

El mandato RCLLNK identifica y, si es posible, corrige los problemas en los sistemas de archivos montados que se están utilizando. El mandato RCLSTG no tiene esta función. No obstante, el mandato RCLSTG puede corregir problemas que el mandato RCLLNK no puede identificar o corregir. En la siguiente tabla se proporciona una comparación más detallada entre los dos mandatos.

Tabla 11. Comparación de los mandatos RCLLNK y RCLSTG

	RCLLNK OBJ('/MiDir/MiObj')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
¿Es necesario que el sistema esté en estado restringido?	No	Sí	No
¿Se pueden utilizar todos los sistemas de archivos durante la operación de reclamación?	Sí	No	Los sistemas de archivos de la ASP independiente que se están reclamando no se pueden utilizar.
¿En qué ASP se pueden reclamar objetos?	Reclama objetos en ASP de sistema, usuario e independientes.	Reclama objetos en ASP de sistema y usuario.	Reclama objetos en ASP independientes.
¿Cómo se reclaman los objetos?	Los objetos se reclaman de forma individual o por subárboles, tal como se especifica en el mandato.	Los objetos se reclaman en todo el sistema.	Los objetos se reclaman por ASP independiente.

Tabla 11. Comparación de los mandatos RCLLNK y RCLSTG (continuación)

	RCLLNK OBJ('/MiDir/MiObj')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
¿Qué problemas de sistemas de archivos conocidos y aplicables se identifican y se corrigen, si es posible?	La mayoría (consulte Reclamar el sistema de archivos ″raíz″ (/), QOpenSys y definido por el usuario para obtener más información).	Todos	Todos
¿Se han encontrado objetos perdidos?	No	Sí	Sí
¿Se reclaman los objetos de sistemas de archivos desmontados?	No	Sí	Sí
¿El mandato admite ejecución en hebras?	Sí	No	No
¿Cuántas instancias del mandato se pueden ejecutar al mismo tiempo?	Varias instancias	Una única instancia	Una única instancia
¿Qué objetos proporcionados por el sistema de archivos integrado aplicable se recrean, si es necesario?	Todos	La mayoría (Consulte Recrear objetos proporcionados por el sistema de archivos integrado para obtener más información).	Ninguno
¿Se pueden identificar los objetos dañados sin reclamarlos?	Sí	No	No

Conceptos relacionados

"Ejemplos: mandato Reclamar enlaces de objetos (RCLLNK)" en la página 111
En estos ejemplos se describen situaciones en las que se puede utilizar el mandato Reclamar enlaces de objetos (RCLLNK) para reclamar objetos en los sistemas de archivos ″raíz″ (/), QOpenSys y en los sistemas de archivos definidos por usuario montados.

Referencia relacionada

"Recrear objetos proporcionados por el sistema de archivos integrado" en la página 111
Esta tabla muestra los objetos proporcionados por el sistema de archivos integrado que vuelve a crear el mandato Reclamar enlaces de objetos (RCLLNK) si no existen. Estos objetos se crean normalmente durante la carga del programa inicial (IPL). Si es necesario, también puede recrear algunos de estos objetos utilizando el mandato Reclamar almacenamiento (RCLSTG).

Información relacionada

Mandato Reclamar almacenamiento (RCLSTG)

Mandato Reclamar enlaces de objetos (RCLLNK)

Mandato Reclamar enlaces de objetos (RCLLNK)

El mandato Reclamar enlaces de objetos (RCLLNK) identifica y repara los objetos dañados en los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario montados sin necesidad de que el sistema esté en un estado restringido. Esto permite corregir los problemas en estos sistemas de archivos sin disminuir la productividad.

En muchos casos, el mandato RCLLNK se puede utilizar como una alternativa al mandato Reclamar almacenamiento (RCLSTG). Por ejemplo, RCLLNK está especialmente indicado para identificar y corregir problemas en los siguientes casos:

- Los problemas están aislados en un único objeto.
- Los problemas están aislados en un grupo de objetos.
- Los objetos dañados se tienen que identificar o suprimir.
- El sistema no puede estar en un estado restringido durante la operación de reclamación.
- Las ASP independientes deben estar disponibles durante la operación de reclamación.

Recrear objetos proporcionados por el sistema de archivos integrado

Esta tabla muestra los objetos proporcionados por el sistema de archivos integrado que vuelve a crear el mandato Reclamar enlaces de objetos (RCLLNK) si no existen. Estos objetos se crean normalmente durante la carga del programa inicial (IPL). Si es necesario, también puede recrear algunos de estos objetos utilizando el mandato Reclamar almacenamiento (RCLSTG).

Tabla 12. Objetos proporcionados por el sistema de archivos integrado y recreados por los mandatos RCLLNK y RCLSTG

Nombre de vía de acceso	Tipo	Recreado por RCLLNK	Recreado por RCLSTG ASPDEV(*SYSBASE)
/dev/zero	*CHRSF	Sí	Sí
/dev/null	*CHRSF	Sí	Sí
/dev/xti/tcp	*CHRSF	Sí	No
/dev/xti/udp	*CHRSF	Sí	No
/etc/vfs	*STMF	Sí	No

Para que el mandato RCLLNK pueda recrear un objeto proporcionado por el sistema de archivos integrado que no existe, se debe ejecutar con el parámetro SUBTREE establecido en *DIR o *ALL cuando se especifica el directorio padre. El mandato debe reclamar de forma satisfactoria el directorio padre del objeto del sistema. Por ejemplo,

```
RCLLNK OBJ('/dev') SUBTREE(*DIR)
```

recrea los objetos *CHRSF /dev/zero y /dev/null si no existen.

Para que el mandato RCLSTG pueda recrear un objeto proporcionado por el sistema de archivos integrado que no existe, se debe ejecutar con el parámetro **ASPDEV** establecido en *SYSBASE y no debe omitirse la parte de recuperación de directorios de la reclamación.

Conceptos relacionados

"Directorios proporcionados" en la página 7
El sistema de archivos integrado crea estos directorios cuando se reinicia el sistema, si todavía no existen.

Referencia relacionada

"Comparación de los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG)" en la página 109
Puede utilizar los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG) para corregir problemas en los sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario.

Información relacionada

Mandato Reclamar enlaces de objetos (RCLLNK)

Ejemplos: mandato Reclamar enlaces de objetos (RCLLNK)

En estos ejemplos se describen situaciones en las que se puede utilizar el mandato Reclamar enlaces de objetos (RCLLNK) para reclamar objetos en los sistemas de archivos "raíz" (/), QOpenSys y en los sistemas de archivos definidos por usuario montados.

Referencia relacionada

"Comparación de los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG)" en la página 109
Puede utilizar los mandatos Reclamar enlaces de objetos (RCLLNK) y Reclamar almacenamiento (RCLSTG) para corregir problemas en los sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario.

Ejemplo: corregir problemas para un objeto

En este caso, los problemas conocidos se aíslan en un objeto. El objeto está dañado y no se puede utilizar, y tampoco se puede restaurar una versión de copia de seguridad del objeto desde el soporte de almacenamiento. Debe corregir el problema rápidamente sin interrumpir las operaciones normales del sistema de archivos.

Para reclamar el objeto, utilice este mandato:

```
RCLLNK
OBJ('/MiDir/MiObjetoAnómalo') SUBTREE(*NONE)
```

donde `/MiDir/MiObjetoAnómalo` es el nombre del objeto dañado que no se puede utilizar.

Ejemplo: corregir problemas que existen en un subárbol de directorios

En este caso, los problemas conocidos se aíslan en un grupo de objetos dentro de un subárbol de directorios. Una aplicación falla debido a los problemas dentro del subárbol de directorios. Debe corregir los problemas rápidamente sin interrumpir las operaciones normales del sistema de archivos.

Para reclamar los objetos dentro del subárbol de directorios, utilice este mandato:

```
RCLLNK OBJ('/DirectorioInstalaciónMiAplicación') SUBTREE(*ALL)
```

donde `DirectorioInstalaciónMiAplicación` es el nombre del directorio que contiene los objetos de problemas.

Ejemplo: buscar todos los objetos dañados en los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario montados

En este caso, una anomalía de disco ha provocado daños en varios objetos. Debe identificar los objetos dañados antes de determinar cómo recuperarlos debidamente.

Necesita una solución que permita identificar los objetos dañados, sin realizar ninguna acción en ellos. No debe interrumpir las operaciones normales del sistema de archivos.

Para identificar los objetos dañados, utilice este mandato:

```
RCLLNK OBJ('/') SUBTREE(*ALL) DMGOBJOPT(*KEEP *KEEP)
```

Asimismo, este mandato también corregirá otros problemas aparte de los objetos dañados cuando identifica los objetos dañados.

Ejemplo: suprimir todos los objetos dañados en los sistemas de archivos "raíz" (/), QOpenSys y definidos por usuario montados

En este caso, una anomalía de disco ha provocado daños en varios objetos. Debe suprimir los objetos dañados para que se pueda restaurar del soporte de almacenamiento una copia de seguridad de los objetos.

Para suprimir los objetos dañados, utilice este mandato:

```
RCLLNK OBJ('/') SUBTREE(*ALL) DMGOBJOPT(*DELETE *DELETE)
```

Los objetos dañados se suprimen sin interrumpir las operaciones normales del sistema de archivos. Asimismo, también se corrigen otros problemas aparte de los objetos dañados cuando estos se suprimen.

Ejemplo: utilizar varios mandatos RCLLNK para reclamar rápidamente todos los objetos de los sistemas de archivos ″raíz″ (/), QOpenSys y definidos por usuario montados

En este caso, como parte del mantenimiento del sistema de rutina, se reclaman todos los objetos de los sistemas de archivos “raíz” (/), QOpenSys y definidos por usuario montados. Asimismo, desea finalizar la operación de reclamación lo más rápido posible para completar el mantenimiento adicional del sistema.

Al dividir la operación de reclamación en grupos independientes, se pueden ejecutar varios mandatos RCLLNK de forma simultánea para que la operación de reclamación termine antes.

Para ejecutar varias operaciones de reclamación en directorios clave del sistema y en otros directorios de nivel superior, utilice los siguientes mandatos (cada uno en una hebra o un trabajo independiente).

```
RCLLNK OBJ('/') SUBTREE(*DIR)
RCLLNK OBJ('/tmp') SUBTREE(*ALL)
RCLLNK OBJ('/home') SUBTREE(*ALL)
RCLLNK OBJ('/etc') SUBTREE(*ALL)
RCLLNK OBJ('/usr') SUBTREE(*ALL)
RCLLNK OBJ('/QIBM') SUBTREE(*ALL)
RCLLNK OBJ('/QOpenSys') SUBTREE(*ALL)
RCLLNK OBJ('/IaspName') SUBTREE(*ALL)
RCLLNK OBJ('/dev') SUBTREE(*ALL)
RCLLNK OBJ('/OtrosDirectoriosNivelSuperior') SUBTREE(*ALL)
```

donde `OtrosDirectoriosNivelSuperior` son los otros directorios que desea reclamar.

Soporte de programación

La adición del sistema de archivos integrado al servidor iSeries en V3R1M0 no afectaba a las aplicaciones de servidor iSeries existentes. Los lenguajes de programación, los programas de utilidad y el soporte del sistema (por ejemplo, las especificaciones de descripción de datos) funcionan del mismo modo que antes de incluir el sistema de archivos integrado.

No obstante, para aprovechar las ventajas de los archivos continuos, directorios y demás soporte del sistema de archivos integrado, se deben utilizan un conjunto de las interfaces de programas de aplicación (API) 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 Family, o las API.

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.

Conceptos relacionados

"Archivo continuo" en la página 18
Un *archivo continuo* es una secuencia de bytes accesible aleatoriamente, sin ninguna otra estructura impuesta por el sistema.

Copiar datos mediante mandatos CL

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

Mandatos CPYTOSTMF y CPYFRMSTMF

Se pueden utilizar los mandatos Copiar desde 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 en archivo de importación (CPYTOIMPF) y Copiar desde 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)
|          FLDDLM(';') RCDDLM(X'07') STRDLM(*DBLQUOTE) 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 FLDDLM, RCDDLM y STRDLM 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.

Información relacionada

Mandato Copiar en archivo continuo (CPYTOSTMF)

Mandato Copiar desde archivo continuo (CPYFRMSTMF)

Mandato Copiar en archivo de importación (CPYTOIMPF)

Mandato Copiar desde archivo de importación (CPYFRMIMPF)

Lenguaje de control (CL)

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 este mismo archivo o en otro archivo.

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Información relacionada

open()

read()

write()

API del sistema de archivos integrado

Copiar datos utilizando funciones de transferencia de datos

Las aplicaciones de transferencia de datos del programa bajo licencia de la iSeries Access Family 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 Family y el uso de recursos del PC y del servidor iSeries, así como las comunicaciones entre ambos.

Si tiene instalado iSeries Access Family 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.

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

Siga estas instrucciones para 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. En la ventana iSeries Access para Windows, pulse **Transferencia de datos desde el servidor iSeries**.

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 al servidor iSeries crea un archivo .DTT o .TFR. La aplicación Transferencia de datos desde el servidor iSeries crea un archivo .DTF o .TTO. En el directorio iSeries Access Family, 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 Pack) para especificar el programa que debe ejecutarse (por ejemplo, `RTOPCB MYFILE.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:

Siga estas instrucciones 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. En la ventana iSeries Access para Windows, pulse **Transferencia de datos al servidor iSeries**.
4. Seleccione el nombre de archivo de PC que desee 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 al servidor iSeries** para ejecutar la transferencia de archivos.

   Nota: Si traslada datos a una definición de base de datos que ya existe en el servidor, la aplicación Transferencia de datos al servidor iSeries 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 el servidor iSeries lo crea cuando transfiere datos de un archivo de base de datos a un archivo continuo. Para completar la transferencia de datos desde 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 al servidor iSeries crea un archivo .DTT o .TFR. La aplicación Transferencia de datos desde el servidor iSeries crea un archivo .DTF o .TTO. En el directorio IBM eServer iSeries Access Family, 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 Pack) para especificar el programa que debe ejecutarse (por ejemplo, `RTOPCB MYFILE.TTO`) y la hora en la que desea que se ejecute el programa.

Referencia relacionada

"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 al servidor iSeries le obligará a utilizar un archivo de descripción de formato (FDF) asociado.

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

Siga las instrucciones 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. En la ventana iSeries Access para Windows, pulse **Transferencia de datos al servidor iSeries**.
4. Abra el menú Herramientas de la aplicación Transferencia de datos al servidor iSeries.
5. Pulse **Crear archivo de base de datos de iSeries**.

   Aparece un asistente que le permitirá crear un archivo de base de datos de iSeries nuevo a partir de un archivo de PC existente. Debe especificar 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 al servidor iSeries 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 el servidor iSeries 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 el servidor iSeries para crear un archivo continuo y su archivo .FDF asociado a partir de este archivo de base de datos.

   Ahora ya puede utilizar la función Transferencia de datos al servidor iSeries. Especifique este archivo .FDF con el archivo continuo fuente que desea transferir.

Referencia relacionada

"Transferir datos de un archivo continuo a un archivo de base de datos" en la página 116
Siga estas instrucciones para transferir datos de un archivo continuo a un archivo de base de datos del servidor.

"Transferir datos de un archivo de base de datos a un archivo continuo" en la página 115
Siga estas instrucciones para para transferir un archivo de un archivo de base de datos a un archivo continuo del servidor.

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 escribirían en cinta o disquete.

El archivo también puede utilizarse como un archivo de base de datos para leer o escribir registros que contengan información de salvar y restaurar. Los archivos de salvar también se pueden utilizar para enviar objetos a otro usuario de la red SNADS.

El mandato Copiar objeto (CPY) puede servir para copiar un archivo de salvar en un archivo continuo o desde él. 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).

Información relacionada

Mandato Copiar objeto (CPY)

Realizar operaciones utilizando interfaces API

Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos 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 mediante Integrated Language Environment (ILE) C/400:

- Funciones C del sistema de archivos integrado que se incluyen en i5/OS.
- Funciones C proporcionadas por el programa bajo licencia ILE C/400.

Para obtener información sobre los programas de salida que admite el sistema de archivos integrado, consulte la Tabla 14 en la página 123.

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

Tabla 13. Interfaces API del sistema de archivos integrado

Función	Descripción
access()	Determinar accesibilidad de archivo
accessx()	Determinar accesibilidad de archivo para una clase de usuarios
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 reescribir archivo existente
creat64()	Crear archivo nuevo o reescribir 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)

Tabla 13. Interfaces API del sistema de archivos integrado (continuación)

Función	Descripción
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
fclear()	Borrar un archivo
fclear64()	Borrar un archivo (con soporte de archivos grandes)
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
getpwnam()	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
ioctl()	Efectuar acción de control de E/S de archivo
link()	Crear enlace con archivo
lseek()	Establecer desplazamiento de lectura/escritura de archivo
lseek64()	Establecer desplazamiento de lectura/escritura 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

Tabla 13. Interfaces API del sistema de archivos integrado (continuación)

Función	Descripción
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 sockets
pread()	Leer en descriptor con desplazamiento
pread64()	Leer en descriptor con desplazamiento (con soporte de archivos voluminosos)
pwrite()	Escribir en descriptor con desplazamiento
pwrite64()	Escribir en descriptor con desplazamiento (con soporte de archivos voluminosos)
QjoEndJournal()	Finalizar registro por diario
QjoRetrieveJournalEntries()	Recuperar entradas de 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)
QlgCreat()	Crear archivo nuevo o reescribir archivo existente (utilizando el nombre de vía de acceso habilitado para NLS)
QlgCreat64()	Crear archivo nuevo o reescribir 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)

Tabla 13. Interfaces API del sistema de archivos integrado (continuación)

Función	Descripción
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)
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)

Tabla 13. Interfaces API del sistema de archivos integrado (continuación)

Función	Descripción
QlgRenameKeep()	Redenominar archivo o directorio, mantener *nuevo* si existe (utilizando el nombre de vía de acceso habilitado para NLS)
QlgRenameUnlink()	Redenominar archivo o directorio, desenlazar *nuevo* 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()	Salvar 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
QP0LCHSG()	Cambiar firma de exploración
Qp0lCvtPathToSYSObjName()	Resolver nombre de vía de acceso del sistema de archivos integrado en el nombre de objeto QSYS
QP0LFLOP()	Realizar operaciones de diversa índole sobre objetos
Qp0lGetAttr()	Obtener los atributos de sistema de un objeto
Qp0lGetPathFromFileID()	Obtener nombre de vía de acceso de objeto a partir de su ID de archivo
Qp0lOpen()	Abrir archivo con nombre de vía de acceso habilitado para NLS
Qp0lProcessSubtree()	Procesar directorios u objetos de un árbol de directorios
Qp0lRenameKeep()	Redenominar archivo o directorio, mantener *nuevo* si existe
Qp0lRenameUnlink()	Redenominar archivo o directorio, desenlazar *nuevo* si existe
QP0LROR()	Recuperar referencias de objetos
QP0LRRO()	Recuperar objetos referenciados
QP0LRTSG()	Recuperar firma de exploración
Qp0lSaveStgFree()	Salvar datos de objetos y liberar su almacenamiento
Qp0lSetAttr()	Establecer atributos de sistema de un objeto

Tabla 13. Interfaces API del sistema de archivos integrado (continuación)

Función	Descripción
Qp0lUnlink()	Eliminar enlace con archivo con nombre de vía de acceso habilitado para NLS
qsysetegid()	Establecer ID de grupo efectivo
qsyseteuid()	Establecer ID de usuario efectivo
qsysetgid()	Establecer ID de grupo
qsysetregid()	Establecer ID de grupo real y efectivo
qsysetreuid()	Establecer ID de usuario real y efectivo
qsysetuid()	Establecer ID de usuario
QZNFRTVE()	Recuperar información de exportación de NFS
read()	Leer en archivo
readdir()	Leer entrada de directorio
readdir_r()	Leer entrada de directorio (permitir ejecución multihebra)
readlink()	Leer valor de enlace simbólico
readv()	Leer en archivo (vector)
rename()	Redenominar archivo o directorio. Se puede definir para tener la semántica de Qp0lRenameKeep() o de Qp0lRenameUnlink().
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()	Escribir en archivo
writev()	Escribir en archivo (vector)

Nota: Algunas de estas funciones también se utilizan en los sockets de i5/OS.

Tabla 14. Programas de salida del sistema de archivos integrado

Función	Descripción
API de exploración del sistema de archivos integrado al cerrar	Se le llama durante el proceso de cierre, por ejemplo, con la API close(). El usuario debe facilitar este programa de salida.

Tabla 14. Programas de salida del sistema de archivos integrado (continuación)

Función	Descripción
API de exploración del sistema de archivos integrado al abrir	Se le llama durante el proceso de apertura, por ejemplo, con la API open(). El usuario debe facilitar este programa de salida.
Procesar un nombre de vía de acceso	Se le llama mediante la API Qp0lProcessSubtree() para cada objeto de la búsqueda de la API que coincida con los criterios de selección del llamador. El usuario debe facilitar este programa de salida.
Salvar con almacenamiento liberado	Se le llama mediante la API Qp0lSaveStgFree() para salvar un objeto de tipo *STMF de iSeries. El usuario debe facilitar este programa de salida.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

Referencia relacionada

"Utilización de API del sistema de archivos integrado en el sistema de archivos ″raíz″ (/)" en la página 35
Todas las API que se listan en el tema Realizar operaciones utilizando API pueden operar en el sistema de archivos "raíz" (/).

"Utilización de las API del sistema de archivos integrado en un sistema de archivos definido por el usuario" en la página 42
Todas las API que se listan en el tema Realizar operaciones utilizando API pueden realizar operaciones en un sistema de archivos definido por el usuario.

"Acceso utilizando API" en la página 93
Puede utilizar interfaces de programa de aplicación (API) para acceder al sistema de archivos integrado.

"Ejemplo: funciones del sistema de archivos integrado C" en la página 130
Este sencillo programa de lenguaje C ilustra la utilización de varias funciones del sistema de archivos integrado.

"Copiar datos mediante API" en la página 115
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 este mismo archivo o en otro archivo.

Información relacionada

Interfaces de programación de aplicaciones (API)

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.

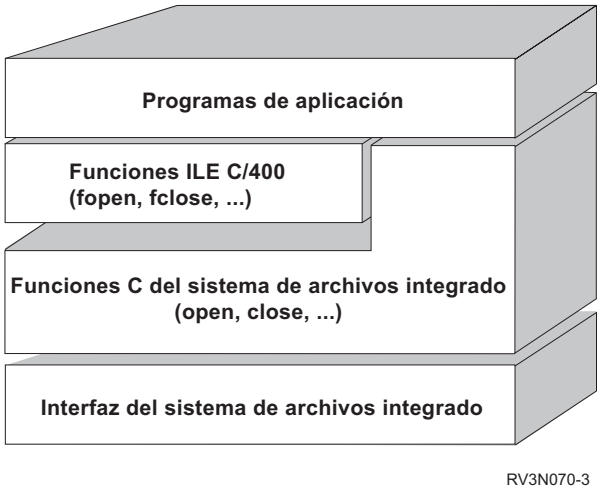

Figura 10. Las funciones ILE C/400 utilizan las funciones de E/S continua del sistema de archivos integrado.

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 C ILE C/400, vea la publicación WebSphere Development Studio: C/C++ Language Reference .

Soporte de archivos voluminosos

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 permite tamaños de archivos continuos de hasta 1 TB (1 TB es igual a aproximadamente 1.099.511.627.776 bytes) en los sistemas de archivos "raíz" (/), QOpenSys y definidos 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.

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

- 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.
- 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.

Información relacionada

API del sistema de archivos integrado

stat64()

stat()

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, este se identifica especificando su vía de acceso de directorio. A continuación se proporciona una visión general 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.
- 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 servidor iSeries, los nombres de vía de acceso no pueden comenzar con un único carácter 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 solo atañe a los nombres de vía de acceso relativa, en los que no hay otros caracteres antes del asterisco (*).
- Cuando se trabaja con objetos del sistema de archivos QSYS.LIB, los nombres de componentes deben tener el formato *nombre.tipo-objeto*; por ejemplo:

  ```
  /QSYS.LIB/PAYROLL.LIB/PAY.FILE
  ```
- Cuando se trabaja con objetos del sistema de archivos QSYS.LIB de ASP independiente, los nombres de componentes deben tener el formato *nombre.tipo-objeto*; por ejemplo:

  ```
  '/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE
  ```
- 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 en los mandatos del sistema de archivos integrado, el asterisco (*), el interrogante (?), las comillas simples (′), las comillas dobles (″) y la tilde (~) carecen de significación especial. Se manejan exactamente igual que otro carácter del nombre. Para evitar confusiones con los valores especiales del servidor iSeries, los nombres de vía de acceso no deben comenzar con un único carácter de asterisco (*). Las únicas API que constituyen una excepción a esta regla son: QjoEndJournal y QjoStartJournal.

- Cuando se utilizan las interfaces de la API Qlg (con nombres de vía de acceso habilitados para NLS), no se permite un valor de carácter nulo como uno de los caracteres del nombre de vía de acceso a menos que se especifique un carácter nulo como delimitador de nombre de vía de acceso.

Conceptos relacionados

"Nombre de vía de acceso" en la página 16
Un *nombre de vía de acceso* le indica al servidor cómo localizar un objeto.

Referencia relacionada

"Reglas que rigen los nombres de vía de acceso en los mandatos CL y pantallas" en la página 77
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.

Información relacionada

API QjoEndJournal

API 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 sockets.

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 solo puede hacer referencia a una descripción de archivo abierto.

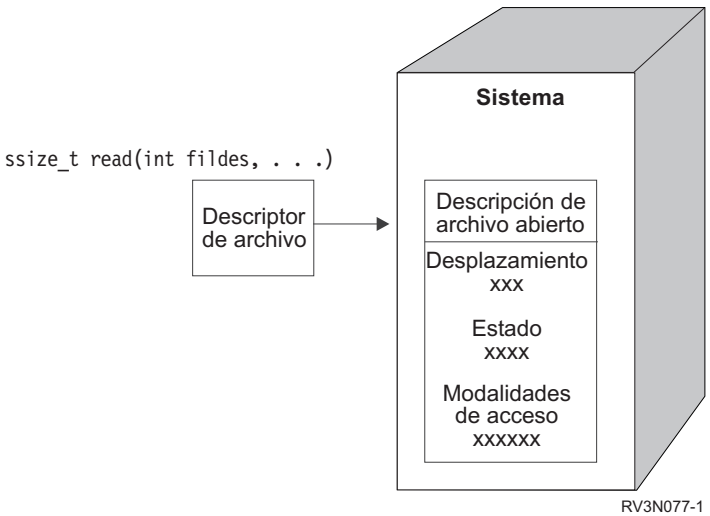

Figura 11. Descriptor de archivo y descripción de archivo abierto

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 definidos por usuario, el acceso a una descripción de archivo abierto se puede pasar de un trabajo a otro, con lo que se permite al trabajo acceder al archivo. Esta operación se efectúa utilizando la función givedescriptor() o takedescriptor() para pasar el descriptor de archivos entre trabajos.

Información relacionada

API givedescriptor()

API takedescriptor()

Programación de sockets

API de sockets

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 para leer y escribir (en el caso de un archivo o un directorio) y para ejecutar (en el caso de un archivo) o buscar (en el caso de 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  .

Información relacionada

API chmod()

API fchmod()

API umask()

API del sistema de archivos integrado

Soporte de sockets

Si la aplicación está utilizando los sistemas de archivos "raíz" (/), QOpenSys o definidos por el usuario, se puede aprovechar el soporte de *socket local* del sistema de archivos integrado. Un objeto socket local (objeto de tipo *SOCKET) 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 socket local. El otro trabajo especifica el nombre del objeto socket local en la función connect(), sendto() o sendmsg().

Una vez establecida la conexión, los dos trabajos pueden enviar datos y recibir datos entre sí utilizando funciones del sistema de archivos integrado como las de write() y read(). Ninguno de los datos que se transfieren pasan en realidad por el objeto socket. El objeto socket es tan solo un punto de encuentro en el que los trabajos se pueden localizar entre sí.

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

El objeto socket local no se puede salvar.

Información relacionada

Programación de sockets

API write()

API read()

API close()

API unlink()

Mandato Eliminar enlace (RMVLNK)

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. 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.

Conceptos relacionados

"Continuidad de nombres" en la página 19
Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definidos por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos.

Conversión de datos

Al acceder a los archivos a través del 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 y escriben en el archivo sin que se conviertan. La aplicación es responsable del manejo de los datos.

Texto Los datos se leen y escriben en el archivo, 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 escriben 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 escribe 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.

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

Información relacionada

API open()

Ejemplo: funciones del sistema de archivos integrado C

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. Establece que el propietario (la persona que ha creado el archivo) tenga autorización de lectura, escritura y ejecución sobre el archivo.

4 Utiliza la función write() para escribir una serie de bytes en el archivo. El descriptor de archivo proporcionado en la operación de apertura (3) identifica el 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, escritura y ejecución en el 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 solo 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 el 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.

Nota: Al utilizar los ejemplos de código, acepta los términos de la "Información de licencia de código y declaración de limitación de responsabilidad" en la página 148.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>


#define BUFFER_SIZE       2048
#define NEW_DIRECTORY     "testdir"
#define TEST_FILE         "test.file"
#define TEST_DATA         "Hello World!"
#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 escribir 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 cambiar de directorio para ir a %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 cambiar de directorio para ir a %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 haberse eliminado 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 escribir, si no existe.
   Otorgar al propietario autorización de lectura, escritura 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
/* Escribir TEST_DATA en TEST_FILE por medio de FilDes */
   BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
   if ( -1 == BytesWritten )
      {
      perror("Error de escritura");
      CleanUpOnError(3);
      return 0;
      }
   printf("Escrito %s en archivo %s.\n",TEST_DATA,TEST_FILE);

 5
/* Cerrar TEST_FILE por medio de FilDes */
   if ( -1 == close(FilDes) )
      {
      perror("Error de cierre");
      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, escritura y ejecución */
   if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )
      {
      perror("Error de 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
   directorio NEW_DIRECTORY recién creado. */
if ( -1 == chdir(NEW_DIRECTORY) )
      {
      perror("Error de 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 de enlace");
      CleanUpOnError(7);
      return 0;
      }
   printf("Creado un enlace %s con %s.\n",LinkName,InitialFile);

 9
/* Abrir el archivo LinkName solo para lectura. */
if ( -1 == (FilDes = open(LinkName,O_RDONLY)) )
      {
      perror("Error de apertura");
      CleanUpOnError(8);
      return 0;
      }
   printf("Abierto %s para lectura.\n",LinkName);

 10
/* Leer en archivo LinkName, por medio de FilDes, en Buffer. */
BytesRead = read(FilDes,Buffer,sizeof(Buffer));
   if ( -1 == BytesRead )
      {
      perror("Error de lectura");
      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 escritos.\n");
      }

 11
/* Cerrar el archivo LinkName por medio de FilDes. */
if ( -1 == close(FilDes) )
      {
      perror("Error de cierre");
      CleanUpOnError(10);
      return 0;
      }
   FilDes = -1;
   printf("Cerrado %s.\n",LinkName);

 12
/* Desenlazar el enlace LinkName con InitialFile. */
if ( -1 == unlink(LinkName) )
      {
```

```
      perror("Error de desenlazar");
      CleanUpOnError(11);
      return 0;
      }
   printf("%s está desenlazado.\n",LinkName);

 13
/* Cambiar el directorio de trabajo actual
   de nuevo por el directorio inicial. */
if ( -1 == chdir(PARENT_DIRECTORY) )
      {
      perror("Error de 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 de 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 de desenlazar");
      CleanUpOnError(14);
      return 0;
      }
   printf("Desenlazando archivo %s.\n",TEST_FILE);


   printf("Programa terminado satisfactoriamente.\n");
   return 0;
}
```

Referencia relacionada

"Realizar operaciones utilizando interfaces API" en la página 118
Muchas de las interfaces de programa de aplicación (API) que realizan operaciones en objetos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Trabajar con archivos y carpetas utilizando iSeries Navigator

Puede realizar estas tareas con los archivos y las carpetas.

Reincorporar un archivo

Siga estos pasos para reincorporar un archivo.

1. En **iSeries Navigator**, pulse con el botón derecho del ratón el archivo que desee reincorporar.
2. Seleccione **Propiedades**.
3. Seleccione **Propiedades de archivo → Utilizar página**.
4. Pulse **Reincorporar**.

Reservar un archivo

Siga estos pasos para reservar un archivo.

1. En **iSeries Navigator**, pulse con el botón derecho del ratón el archivo que desee reservar.

2. Seleccione **Propiedades**.
3. Pulse **Propiedades de archivo → Utilizar página**.
4. Pulse **Reservar**.

Crear una carpeta

Siga estos pasos para crear una carpeta.

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

Cuando se crea una carpeta en el servidor iSeries, debe considerar si quiere proteger la carpeta (u objeto) nuevo con la administración de diario. También debe considerar si desea que se exploren los objetos que se creen en esta carpeta.

Tareas relacionadas

"Establecer si debe o no explorarse los objetos" en la página 141
Siga estos pasos para establecer si un objeto debe o no explorarse.

Información relacionada

Gestión por diario

Eliminar una carpeta

Siga estos pasos para eliminar una carpeta.

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

Trasladar archivos o carpetas 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.

También deberá tener en cuenta la siguiente situación:

- ¿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 dan soporte a los mandatos y las API del sistema de archivos jerárquico (HFS) para trabajar con los objetos de documento y carpeta. 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 solo 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.

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.

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.

Conceptos relacionados

"Trabajar con sistemas de archivos" en la página 27
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.

Referencia relacionada

"Comparación de los sistemas de archivos" en la página 29
En estas tablas se resumen las características y las limitaciones de cada sistema de archivos.

Información relacionada

Mandato Crear directorio (CRTDIR)

Mandato Mover (MOV)

Mandato Copiar (CPY)

Establecer permisos

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 solo puedan ver los objetos, mientras que a otros se les puede dejar editarlos.

Para establecer los permisos para un archivo o carpeta, siga estos pasos:

1. Expanda el sistema que desee utilizar en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado**. Continúe desplegando opciones hasta que vea el objeto para el que se quiere añadir permisos.
2. Pulse con el botón derecho el objeto para el que se quiere añadir permisos y seleccione **Permisos**.
3. Pulse **Añadir** en el recuadro de diálogo **Permisos**.
4. 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**.
5. Pulse **Aceptar**. Esto añadirá los usuarios o grupos al inicio de la lista.
6. Pulse el botón **Detalles** para implementar permisos más detallados.
7. Aplique los permisos que desea para el usuario mediante el recuadro de selección adecuado.
8. Pulse **Aceptar**.

Configurar la conversión de archivos de texto

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, siga estos pasos:

1. Expanda el sistema que desea utilizar en **iSeries Navigator** → **Sistemas de archivos**.
2. Pulse con el botón derecho del ratón **Sistema de archivos integrado** y seleccione **Propiedades**.
3. 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**.
4. Repita el paso 3 para todas las extensiones de archivo que quiera convertir automáticamente.
5. Pulse **Aceptar**.

Enviar un archivo o carpeta a otro sistema

Siga estos pasos para enviar un archivo o carpeta a otro sistema.

1. Expanda el sistema que desea utilizar en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado**. Continúe desplegando opciones hasta que vea el archivo o carpeta que se quiere enviar.
2. Pulse con el botón derecho del ratón el archivo o la carpeta y seleccione **Enviar**. El archivo o carpeta aparecerá en la lista Archivos y carpetas seleccionados del recuadro de diálogo Enviar archivos desde.
3. Despliegue la lista de equipos y grupos disponibles.
4. 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.
5. Pulse **Aceptar** para enviar el archivo o carpeta con las definiciones del paquete por omisión y la información de planificación actuales.

Cuando se crea una definición de paquete, esta se salva 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.

Tareas relacionadas

"Cambiar opciones de una definición del paquete"
Una definición de paquete le permite agrupar un conjunto de objetos de i5/OS o archivos del sistema de archivos integrado.

"Planificar fecha y hora en que enviar el archivo o carpeta"
La función de planificación le ofrece la flexibilidad de realizar su trabajo cuando le resulte más conveniente.

Cambiar opciones de una definición del paquete

Una definición de paquete le permite agrupar un conjunto de objetos de i5/OS 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, siga estos pasos:

1. Lleve a cabo todos los pasos de "Enviar un archivo o carpeta a otro sistema" en la página 138.
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.

Tareas relacionadas

"Enviar un archivo o carpeta a otro sistema" en la página 138
Siga estos pasos para enviar un archivo o carpeta a otro sistema.

"Planificar fecha y hora en que enviar el archivo o carpeta"
La función de planificación le ofrece la flexibilidad de realizar su trabajo cuando le resulte más conveniente.

Planificar fecha y hora en que enviar el archivo o carpeta

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 138.
2. Pulse **Planificar**.
3. Seleccione las opciones relativas al momento en que se quiere enviar el archivo o carpeta.

Tareas relacionadas

"Enviar un archivo o carpeta a otro sistema" en la página 138
Siga estos pasos para enviar un archivo o carpeta a otro sistema.

"Cambiar opciones de una definición del paquete"
Una definición de paquete le permite agrupar un conjunto de objetos de i5/OS o archivos del sistema de archivos integrado.

Crear un compartimiento de archivos

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

Para crear un compartimiento de archivos, siga estos pasos.

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

Cambiar un compartimiento de archivos

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

Para cambiar un compartimiento de archivos, siga estos pasos.

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

Crear un nuevo sistema de archivos definido por el usuario

Sistema de archivos definido por usuario (UDFS) es un sistema de archivos creado por el usuario y cuyos atributos también define el usuario. Los UDFS residen en agrupaciones de almacenamiento auxiliar (ASP) en el sistema.

Para crear un nuevo sistema de archivos definido por el usuario (UDFS), siga estos pasos:

1. Expanda el sistema en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado** → **Raíz** → **Dev**.
2. Pulse la agrupación de almacenamiento auxiliar (ASP) que desea que contenga el nuevo UDFS.
3. Seleccione **Nuevo UDFS** en el menú Archivo.
4. Especifique el nombre del UDFS, una descripción (opcional), los valores de auditoría, el atributo de exploración por omisión y si los archivos en el nuevo UDFS tendrán nombres de archivo sensibles a las mayúsculas y minúsculas en el diálogo Nuevo sistema de archivos definido por el usuario.

Montar un sistema de archivos definido por el usuario

Para acceder o visualizar los datos almacenados en un UDFS, debe montar el UDFS después de cada IPL.

Cuando se monta un UDFS, cubre todos los sistemas de archivos, directorios u objetos que existen debajo del punto de montaje en la jerarquía de carpetas. Esto hace que dichos sistemas de archivos, directorios u objetos sean inaccesibles hasta que se desmonta el UDFS. Para garantizar que se mantenga el acceso a todos los datos del sistema de archivos integrado, monte el UDFS en una carpeta vacía. Una vez montado el UDFS, se podrá acceder a los archivos del UDFS desde esa carpeta. Los cambios que se hagan en la carpeta serán cambios realizados 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), siga estos pasos:

1. Expanda el sistema en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado** → **Raíz** → **Dev**.
2. Pulse la agrupación de almacenamiento auxiliar (ASP) que contiene el UDFS que desea montar.
3. Pulse con el botón derecho el UDFS que desea montar en la columna **Nombre de UDFS** del panel derecho de Operations Navigator.
4. 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

Cuando se monta un UDFS, cubre todos los sistemas de archivos, directorios u objetos que existen debajo del punto de montaje en la jerarquía de carpetas. Esto hace que dichos sistemas de archivos, directorios u objetos sean inaccesibles hasta que se desmonta el UDFS.

Para desmontar un sistema de archivos definido por el usuario (UDFS), siga estos pasos:

1. Expanda el sistema en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado** → **Raíz** → **Dev**.
2. Pulse la agrupación de almacenamiento auxiliar (ASP) que contiene el UDFS que desea desmontar.
3. Pulse con el botón derecho el UDFS que desea desmontar en la columna **Nombre de UDFS** del panel derecho de iSeries Navigator.
4. Seleccione **Desmontar**.

Establecer si debe o no explorarse los objetos

Siga estos pasos para establecer si un objeto debe o no explorarse.

1. Despliegue el sistema en **iSeries Navigator** → **Sistemas de archivos** → **Sistema de archivos integrado**.
2. Expanda la carpeta o archivo en cuestión.
3. Pulse con el botón derecho sobre la carpeta o archivo y seleccione **Propiedades**
4. Seleccione la pestaña **Seguridad**.
5. Seleccione **Explorar objetos** con la opción que desee.

Para obtener más información sobre las opciones, consulte los siguientes apartados. Las descripciones para estas opciones son para archivos. Solo es posible explorar archivos. Con carpetas y sistemas de archivos definidos por el usuario, puede especificarse qué atributo de exploración debería concederse a los archivos creados en esa carpeta o sistema de archivos definido por el usuario.

- Sí

  El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración si el objeto ha sido modificado o si el software de exploración ha sido actualizado desde la última vez en que se exploró el objeto.
- No

  Los programas de salida relacionados con la exploración no explorarán el objeto.

  Nota: Si en los valores del sistema se selecciona la opción `Explorar en el siguiente acceso tras la restauración del objeto`, cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración.
- Solo cuando el objeto ha cambiado

El objeto se explorará según las reglas descritas en los programas de salida relacionados con la exploración solo si el objeto ha sido modificado desde la última vez en que se exploró. No se explorará si el software de exploración ha sido actualizado.

Si la opción `Utilizar el atributo solo cuando haya objetos cambiados para controlar la exploración` no se especifica en los valores del sistema, no se utilizará este atributo `solo cambio de objeto` y el objeto se explorará tras haber sido modificado y cuando el software de exploración indique una actualización.

Notas:

1. En esta pestaña de archivos, también puede determinar el estado de exploración de un objeto.
2. Si en los valores del sistema se selecciona la opción `Explorar en el siguiente acceso tras la restauración del objeto`, cuando se restaura un objeto con este atributo, el objeto se explorará como mínimo una vez tras la restauración.

Conceptos relacionados

"Soporte para la exploración" en la página 21
Con iSeries, puede explorar objetos del sistema de archivos integrado.

Tareas relacionadas

"Crear una carpeta" en la página 136
Siga estos pasos para crear una carpeta.

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.

RPC incluye un estándar para la representación de datos, llamado eXternal Data Representation (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 que actualmente están 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.

API de selección 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
freenetconfigent()	Libera la estructura netconfig devuelta a partir de la llamada a la función getnetconfigent()
getnetconfig()	Devuelve el puntero al registro actual del archivo netconfig e incrementa su puntero hasta el registro siguiente
getnetconfigent()	Devuelve el puntero a la estructura netconfig que corresponde al identificador de red de entrada

API	Descripción
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.

Información relacionada

Buscador de API

API de conversión de nombre a dirección

Estas API permiten 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
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_sperror()	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)

Información relacionada

Buscador de API

API XDR (eXternal Data Representation)

Estas API permiten que las aplicaciones 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.

API	Descripción
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.
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

API	Descripción
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.

Información relacionada

Buscador de API

API de autenticación

Estas API proporciona autenticación en las aplicaciones 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

Información relacionada

Buscador de API

Interfaces API TI-RPC (RPC independiente del transporte)

Estas API 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.

Información relacionada

Buscador de API

Interfaces API TI-RPC simplificadas

Estas API simplificadas 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

Información relacionada

Buscador de API

Interfaces API TI-RPC de nivel superior

Estas API permiten 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 de cliente genérico
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

Información relacionada

Buscador de API

Interfaces API TI-RPC de nivel intermedio

Estas API 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

Información relacionada

Buscador de API

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
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()

Información relacionada

Buscador de API

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

Información relacionada

Buscador de API

Información relacionada con el sistema de archivos integrado

A continuación se describen los manuales de productos, sitios Web y temas de Information Center relacionados con el tema del sistema de archivos integrado. Puede ver o imprimir los PDF.

Manuales

- Soporte del sistema de archivos de red de i5/OS En esta publicación se describe el sistema de archivos de red por medio de una serie de aplicaciones procedentes 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 Esta publicación sirve de guía del usuario y manual de consulta para IBM Optical Support en i5/OS. 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: C/C++ Language Reference Esta publicación proporciona la información necesaria para diseñar, editar, compilar, ejecutar y depurar programas ILE C/400 en el servidor iSeries.
- Security — Reference Esta publicación proporciona información técnica detallada sobre seguridad de i5/OS, incluidos los valores del sistema relacionados con la seguridad que influyen en el proceso relacionado con la exploración del sistema de archivos integrado.
- APPC Programming Esta publicación 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.
- Backup and Recovery Esta publicación proporciona información general sobre las opciones de recuperación y disponibilidad para el servidor IBM iSeries.

Más información

- **Informes de experiencias**

Los informes de experiencias están redactados por los programadores de IBM y documentan sus experiencias prácticas durante la implementación de soluciones reales en situaciones reales. Utilícelos para conocer las experiencias de los programadores de IBM con una implementación específica de una solución iSeries, con consejos e instrucciones detalladas. El informe de experiencia Salvar el sistema de archivos integrado está relacionado con archivos y sistemas de archivos.

- Lenguaje de control
- Globalización de i5/OS
- Interfaces de programación de aplicaciones (API)
- Gestión por diario
- Control de compromiso

Cómo guardar los archivos PDF

Si desea guardar un archivo PDF en su estación de trabajo para verlo o imprimirlo:

1. Pulse el PDF con el botón derecho del ratón en el navegador (pulse el enlace anterior con el botón derecho del ratón).
2. Pulse la opción que guarda el PDF localmente.
3. Navegue hasta el directorio en el que desea guardar el PDF.
4. Pulse **Guardar**.

Cómo descargar Adobe Reader

Para poder ver o imprimir archivos PDF, debe instalar Adobe Reader en su sistema. Puede descargar una copia gratuita desde el sitio Web de Adobe (www.adobe.com/products/acrobat/readstep.html) .

Información de licencia de código y declaración de limitación de responsabilidad

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

SUJETO A LAS GARANTÍAS ESTATUTARIAS QUE NO PUEDAN EXCLUIRSE, IBM, LOS DESARROLLADORES Y LOS SUMINISTRADORES DE PROGRAMAS NO OFRECEN NINGUNA GARANTÍA NI CONDICIÓN, YA SEA IMPLÍCITA O EXPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS O CONDICIONES IMPLÍCITAS DE COMERCIALIZACIÓN, ADECUACIÓN A UN PROPÓSITO DETERMINADO Y NO VULNERACIÓN CON RESPECTO AL PROGRAMA O AL SOPORTE TÉCNICO, SI EXISTE.

BAJO NINGUNA CIRCUNSTANCIA, IBM, LOS DESARROLLADORES O SUMINISTRADORES DE PROGRAMAS SE HACEN RESPONSABLES DE NINGUNA DE LAS SIGUIENTES SITUACIONES, NI SIQUIERA EN CASO DE HABER SIDO INFORMADOS DE TAL POSIBILIDAD:

1. PÉRDIDA O DAÑOS CAUSADOS EN LOS DATOS;
2. DAÑOS ESPECIALES, ACCIDENTALES, DIRECTOS O INDIRECTOS, O DAÑOS ECONÓMICOS DERIVADOS;
3. PÉRDIDAS DE BENEFICIOS, COMERCIALES, DE INGRESOS, CLIENTELA O AHORROS ANTICIPADOS.

ALGUNAS JURISDICCIONES NO PERMITEN LA EXCLUSIÓN O LA LIMITACIÓN DE LOS DAÑOS DIRECTOS, ACCIDENTALES O DERIVADOS, POR LO QUE PARTE DE LAS LIMITACIONES O EXCLUSIONES ANTERIORES, O TODAS ELLAS, PUEDE NO SER PROCEDENTE EN SU CASO.

Apéndice. Notas

Esta información se ha escrito para productos y servicios ofrecidos en Estados Unidos de América.

Es posible que en otros países IBM no ofrezca los productos, los servicios o las características que se describen en este documento. El representante local de IBM le puede informar acerca de los productos y servicios que actualmente están disponibles en su localidad. Las referencias hechas a productos, programas o servicios de IBM no pretenden afirmar ni dar a entender que únicamente puedan utilizarse dichos productos, programas o servicios de IBM. Puede utilizarse en su lugar cualquier otro producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes de aprobación que cubran los temas descritos en este documento. La posesión de este documento no le otorga licencia sobre dichas patentes. Puede enviar las consultas sobre licencias, por escrito, a la siguiente dirección:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
Estados Unidos

Para consultas sobre licencias relativas a la información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM en su país o envíe las consultas, por escrito, a:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japón

El párrafo siguiente no es de aplicación en el Reino Unido ni en ningún otro país en el que tales disposiciones sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN “TAL CUAL” SIN GARANTÍA DE NINGUNA CLASE, EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN Y DE COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de responsabilidad, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información incluida en este documento está sujeta a cambios periódicos, que se incorporarán en nuevas ediciones de la publicación. IBM puede efectuar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan únicamente por cortesía y de ningún modo deben interpretarse como promoción de dichos sitios Web. Los materiales de estos sitios Web no forman parte de los materiales de IBM para este producto, y el usuario será responsable del uso que se haga de estos sitios Web.

IBM puede utilizar o distribuir la información que usted le suministre del modo que IBM considere conveniente sin incurrir por ello en ninguna obligación para con usted.

Los licenciatarios de este programa que deseen obtener información acerca del mismo con el fin de: (i) intercambiar la información entre programas creados independientemente y otros programas (incluido este) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
Estados Unidos

Esta información puede estar disponible, sujeta a los términos y condiciones pertinentes, e incluir en algunos casos el pago de una cantidad.

El programa bajo licencia descrito en esta información, así como todo el material bajo licencia disponible para él, lo proporciona IBM bajo los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas bajo Licencia de IBM, el Acuerdo de Licencia para Código Máquina de IBM o cualquier otro acuerdo equivalente entre ambas partes.

Los datos de rendimiento incluidos aquí se determinaron en un entorno controlado. Por lo tanto, los resultados que se obtengan en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas que estén en fase de desarrollo y no existe ninguna garantía de que esas mediciones vayan a ser iguales en los sistemas disponibles en el mercado. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información concerniente a productos que no son de IBM se ha obtenido de los suministradores de dichos productos, de sus anuncios publicados o de otras fuentes de información pública disponibles. IBM no ha comprobado dichos productos y no puede afirmar la exactitud en cuanto a rendimiento, compatibilidad u otras características relativas a productos no IBM. Las consultas acerca de las prestaciones de los productos que no son de IBM deben dirigirse a los suministradores de tales productos.

Todas las declaraciones relativas a la dirección o intención futura de IBM están sujetas a cambios o anulación sin previo aviso y representan únicamente metas y objetivos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlas de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y direcciones utilizados por una empresa real es mera coincidencia.

LICENCIA DE DERECHOS DE COPIA:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir los programas de ejemplo de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, la facilidad de mantenimiento ni el funcionamiento de los programas.

Cada copia o parte de estos programas de ejemplo, así como todo trabajo derivado, debe incluir un aviso de copyright como el siguiente:

© (nombre de su empresa) (año). Algunas partes de este código se derivan de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _escriba el año o los años_. Reservados todos los derechos.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Información sobre la interfaz de programación

Esta publicación del sistema de archivos integrado documenta las interfaces de programación cuya finalidad es permitir al cliente escribir programas para obtener los servicios de IBM i5/OS.

Marcas registradas

Los términos siguientes son marcas registradas de International Business Machines Corporation en Estados Unidos y/o en otros países:

| C/400
| DB2
| i5/OS
| IBM
| IBM (logotipo)
| Integrated Language Environment
| iSeries
| NetServer
| OfficeVision
| OS/2
| OS/400
| WebSphere
| xSeries

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o en otros países.

| Linux es una marca registrada de Linus Torvalds en Estados Unidos y/o en otros países.

UNIX es una marca registrada de The Open Group en Estados Unidos y en otros países.

Los demás nombres de compañías, productos y servicios pueden ser marcas registradas o de servicio de terceros.

Términos y condiciones

Los permisos para utilizar estas publicaciones están sujetos a los siguientes términos y condiciones.

Uso personal: puede reproducir estas publicaciones para uso personal (no comercial) siempre y cuando incluya una copia de todos los avisos de derechos de autor. No puede distribuir ni visualizar estas publicaciones ni ninguna de sus partes, como tampoco elaborar trabajos que se deriven de ellas, sin el consentimiento explícito de IBM.

Uso comercial: puede reproducir, distribuir y visualizar estas publicaciones únicamente dentro de su empresa, siempre y cuando incluya una copia de todos los avisos de derechos de autor. No puede elaborar trabajos que se deriven de estas publicaciones, ni tampoco reproducir, distribuir ni visualizar estas publicaciones ni ninguna de sus partes fuera de su empresa, sin el consentimiento explícito de IBM.

Aparte de la autorización que se concede explícitamente en este permiso, no se otorga ningún otro permiso, licencia ni derecho, ya sea explícito o implícito, sobre las publicaciones, la información, los datos, el software o cualquier otra propiedad intelectual contenida en ellas.

IBM se reserva el derecho de retirar los permisos aquí concedidos siempre que, según el parecer de IBM, las publicaciones se utilicen en detrimento de sus intereses o cuando, también según el parecer de IBM, no se sigan debidamente las instrucciones anteriores.

No puede bajar, exportar ni reexportar esta información si no lo hace en plena conformidad con la legislación y normativa vigente, incluidas todas las leyes y normas de exportación de Estados Unidos.

IBM NO PROPORCIONA NINGUNA GARANTÍA SOBRE EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, NO VULNERACIÓN E IDONEIDAD PARA UN FIN DETERMINADO.

Impreso en España