



@server

iSeries

Sistema de Ficheiros Integrado

*Versão 5 Edição 3*







@server

iSeries

Sistema de Ficheiros Integrado

*Versão 5 Edição 3*

**Nota**

Antes de utilizar as informações contidas neste manual, bem como o produto a que elas se referem, não deixe de ler as “Informações especiais”, na página 127.

**Sexta edição (Agosto de 2005)**

Esta edição aplica-se à versão 5, edição 3, modificação 0 do Operating System/400 da IBM (número de produto 5722-SS1) e a todas as edições e modificações subsequentes, salvo indicação em contrário nas novas edições. Não é possível executar esta versão em todos os modelos de computador de conjunto de instruções reduzido (RISC - Reduced Instruction Set Computer), nem em modelos de computador de conjunto de instruções complexo (CICS® - Complex Instruction Set Computer).

© Copyright International Business Machines Corporation 1999, 2005. Todos os direitos reservados.

# Índice

## Sistema de ficheiros integrado . . . . . 1

Informações de exclusão de responsabilidade para códigos . . . . .	2
O que há de novo na V5R3 . . . . .	2
Imprimir este tópico . . . . .	3
Descrição geral do sistema de ficheiros integrado . . . . .	3
O que é o sistema de ficheiros integrado . . . . .	3
Porquê utilizar o sistema de ficheiros integrado? . . . . .	4
Conceitos . . . . .	5
Directório . . . . .	5
Ligação . . . . .	11
Nome de caminho . . . . .	15
Ficheiro de dados contínuos . . . . .	15
Continuidade de nomes . . . . .	17
Atributos expandidos . . . . .	17
Suporte de pesquisa . . . . .	18
Trabalhar com sistemas de ficheiros . . . . .	23
Comparação entre sistemas de ficheiros . . . . .	25
Sistema de ficheiros "raiz" (/) . . . . .	28
Sistema de ficheiros de sistemas abertos (QOpenSys) . . . . .	30
Sistema de ficheiros definido pelo utilizador (UDFS) . . . . .	32
Sistema de ficheiros de biblioteca (QSYS.LIB) . . . . .	37
QSYS.LIB de ASP independente . . . . .	40
Sistema de ficheiros de serviços da biblioteca de documentos (QDLS) . . . . .	43
Sistema de Ficheiros Óptico (QOPT) . . . . .	45
Sistema de ficheiros NetWare (QNetWare) . . . . .	47
Sistema de ficheiros do iSeries NetClient (QNTC) . . . . .	50
Sistema de ficheiros do Servidor de Ficheiros do OS/400 (QFileSvr.400) . . . . .	53
Sistema de ficheiros da rede (NFS) . . . . .	57
Aceder ao sistema de ficheiros integrado . . . . .	60
Aceder através de menus e ecrãs . . . . .	61
Aceder através de comandos de CL . . . . .	62
Aceder utilizando APIs . . . . .	80
Aceder utilizando o iSeries Navigator . . . . .	80
Aceder utilizando o iSeries NetServer . . . . .	81
Aceder utilizando programas de transferência de ficheiros . . . . .	82
Aceder utilizando um PC . . . . .	83
Converter directórios de *TYPE1 para *TYPE2 . . . . .	84
Descrição geral da conversão de *TYPE1 para *TYPE2 . . . . .	84
Sugestões: Conversões . . . . .	84
Registar objectos em diário . . . . .	88

Descrição geral do registo em diário . . . . .	88
Iniciar registo em diário . . . . .	93
Alterar registo em diário . . . . .	93
Terminar registo em diário . . . . .	93
Suporte de programação . . . . .	93
Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados . . . . .	94
Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda . . . . .	98
Executar operações utilizando APIs . . . . .	98
Suporte de sockets . . . . .	108
Nomenclatura e suporte internacional . . . . .	108
Conversão de dados . . . . .	108
Exemplo: Funções C do Sistema de ficheiros integrado . . . . .	109
Trabalhar com ficheiros e pastas através da utilização do iSeries Navigator . . . . .	114
Dar entrada de um ficheiro . . . . .	115
Dar saída de um ficheiro . . . . .	115
Criar uma pasta . . . . .	115
Remover uma pasta . . . . .	116
Mover ficheiros ou pastas para outro sistema de ficheiros . . . . .	116
Definir permissões . . . . .	117
Configurar conversão de texto do ficheiro . . . . .	117
Enviar um ficheiro ou uma pasta para outro sistema . . . . .	118
Alterar opções para uma definição do pacote . . . . .	118
Marcar uma data e uma hora para enviar o ficheiro ou a pasta . . . . .	118
Criar uma partilha de ficheiros . . . . .	119
Alterar uma partilha de ficheiros . . . . .	119
Definir se os objectos devem ou não ser pesquisados . . . . .	119
Informações relacionadas com o sistema de ficheiros integrado . . . . .	120
Chamada de Procedimento Remoto . . . . .	120
Independente de Transporte . . . . .	120
Referências . . . . .	125
Relatórios de Experiências . . . . .	126

## Apêndice. Informações especiais . . . . . 127

Informações sobre interfaces de programação . . . . .	129
Marcas comerciais . . . . .	129
Termos e condições para descarregamento e impressão de publicações . . . . .	129



---

## Sistema de ficheiros integrado

O sistema de ficheiros integrado é uma parte do OS/400<sup>®</sup> que suporta input/output de sequência e gestão de memória semelhante ao computador pessoal e aos sistemas operativos UNIX<sup>®</sup>, fornecendo ao mesmo tempo uma estrutura de integração em todas as informações armazenadas no servidor.

Consulte os tópicos que se seguem para obter informações detalhadas sobre o sistema de ficheiros integrado no OS/400.

### **O que há de novo na V5R3**

Verifique como este tópico foi alterado desde a última edição.

### **PDF pronto para impressão**

Aprenda a apresentar ou imprimir uma versão PDF deste tópico.

### **Descrição geral do sistema de ficheiros integrado**

Saiba quais os princípios básicos do sistema de ficheiros integrado.

### **Conceitos**

Faça aqui a sua pesquisa para obter informações sobre os diferentes conceitos relativos ao sistema de ficheiros integrado.

### **Trabalhar com sistemas de ficheiros**

Aprenda a gerir os diferentes sistemas de ficheiros que existem no OS/400.

### **Aceder ao sistema de ficheiros integrado**

Conheça as diferentes opções de acesso ao sistema de ficheiros integrado. As opções incluem: utilização de menus e ecrãs, comandos de CL, APIs, iSeries<sup>™</sup> NetServer, iSeries Navigator e programas de transferência de ficheiros.

### **Converter Directórios de \*TYPE1 para \*TYPE2**

Saiba como o sistema converte automaticamente directórios \*TYPE1 para directórios \*TYPE2.

### **Registar objectos em diário**

Saiba como iniciar, terminar e alterar o registo em diário de objectos do sistema de ficheiros integrado.

### **Suporte de programação**

Saiba como copiar dados, utilizar APIs e executar outras funções de suporte.

### **Trabalhar com ficheiros e pastas através da utilização do iSeries Navigator**

Faça aqui a sua pesquisa para saber como executar tarefas com ficheiros e pastas, tais como, dar entrada e saída de um ficheiro e criar ou remover uma pasta.

### **Informações relacionadas com o sistema de ficheiros integrado**

Encontre informações no Centro de Informações do iSeries e na Internet com tópicos adicionais sobre o sistema de ficheiros integrado.

**Nota:** Consulte “Informações de exclusão de responsabilidade para códigos” na página 2 para obter informações legais importantes.

---

## Informações de exclusão de responsabilidade para códigos

Este documento contém exemplos de programação.

A IBM® concede uma licença de direitos de não exclusiva para utilizar todos os exemplos de código de programação dos quais pode gerar funções semelhantes personalizadas de acordo com as suas necessidades específicas.

Todos os códigos de exemplo são fornecidos pela IBM apenas para fins ilustrativos. Estes exemplos não foram testados exaustivamente sob todas as condições. Deste modo, a IBM não garante nem se responsabiliza pela fiabilidade, assistência ou funcionamento destes programas.

Todos os programas aqui incluídos são fornecidos "COMO ESTÃO", sem quaisquer garantias. São expressamente excluídas as garantias implícitas de não infracção, comercialização e adequação a um determinado fim.

---

## O que há de novo na V5R3

As informações que se seguem foram adicionadas ou actualizadas nesta edição:

### APIs do sistema de ficheiros integrado

O sistema de ficheiros integrado adicionou funções e suporte às seguintes APIs:

- Eliminar um ficheiro (fclear())
- Eliminar um ficheiro (suporte de ficheiros grandes) (fclear64())
- Alterar Assinatura de Pesquisa (QP0LCHSG)
- Obter Objectos Referenciados (QP0LRRO)
- Obter Assinatura de Pesquisa (QP0LRTSG)

### Comandos de CL do sistema de ficheiros integrado

Na V5R3, o sistema de ficheiros integrado adicionou funções e suporte aos seguintes comandos de CL:

- Comando Alterar Objecto Registado em Diário (CHGJRNOBJ)
- Comando Imprimir Informações do Directório (PRTDIRINF)
- Comando Obter Informações do Directório (RTVDIRINF)

Para obter informações sobre o output dos comandos PRTDIRINF e RTVDIRINF, consulte "Trabalhar com output dos comandos RTVDIRINF e PRTDIRINF" na página 67.

### Dois controladores de dispositivos

O sistema de ficheiros "raiz" (/) no directório /dev/xti não suportará dois controladores de dispositivos com o nome udp e tcp. O tópico que se segue aborda os controladores de dispositivos:

- "Dispositivos UDP e TCP no sistema de ficheiros "raiz" (/)" na página 30

### Configuração de Kerberos para sistemas de ficheiros QNTC

Em edições anteriores, o QNTC utilizava uma palavra-passe para proceder à respectiva autenticação perante os servidores remotos. Para simplificar a administração de palavras-passe, o QNTC começará por utilizar uma senha de servidor kerberos para a autenticação com os servidores remotos. O tópico "Activar Kerberos para utilização com o sistema de ficheiros QNTC" na página 53 dispõe de mais informações sobre como configurar kerberos para ser utilizado com o sistema de ficheiros QNTC.

### Conversão automática para directórios \*TYPE2

Pouco tempo depois da instalação da V5R3, o sistema inicia automaticamente a conversão de quaisquer sistemas de ficheiros que suportem actualmente directórios \*TYPE1 para directórios \*TYPE2. Consulte “Converter directórios de \*TYPE1 para \*TYPE2” na página 84 para obter mais informações.

## Suporte de pesquisa

Actualmente, o iSeries tem a possibilidade de permitir a pesquisa de objectos do sistema de ficheiros integrado. As pesquisas podem ser efectuadas durante operações de abertura ou encerramento dos objectos. Os objectos podem ser pesquisados por vários motivos, tais como, a verificação da existência de vírus. Também pode definir quando pretende que sejam efectuadas as pesquisas, conforme as suas necessidades. O tópico que se segue fornece informações adicionais sobre suporte de pesquisa e o que é necessário para o activar:

- “Suporte de pesquisa” na página 18

---

## Imprimir este tópico

Para visualizar ou descarregar a versão PDF deste documento, consulte o tópico “Sistema de ficheiros integrado”  (cerca de 1.455 KB).

## Guardar ficheiros PDF

Para guardar um PDF na estação de trabalho para visualização ou impressão:

1. No browser, faça clique com o botão direito do rato no PDF pretendido (faça clique com o botão direito do rato na ligação acima).
2. Faça clique em **Guardar Destino Como...** se estiver a utilizar o Internet Explorer. Faça clique em **Save Link As...** (Guardar ligação como...) se estiver a utilizar o Netscape Communicator.
3. Navegue para o directório no qual pretende guardar o PDF.
4. Faça clique em **Save** (Guardar).

## Descarregar o Adobe Acrobat Reader

Necessita de ter o Adobe Acrobat Reader para visualizar ou imprimir estes PDFs. Pode descarregar uma cópia a partir do sítio da Web Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)). 

---

## Descrição geral do sistema de ficheiros integrado

Os seguintes tópicos descrevem o sistema de ficheiros integrado no servidor iSeries e mostram como pode ser utilizado no seu servidor.

- “O que é o sistema de ficheiros integrado”
- “Porquê utilizar o sistema de ficheiros integrado?” na página 4

## O que é o sistema de ficheiros integrado

O **sistema de ficheiros integrado** é uma parte do OS/400 que suporta input/output de sequência e gestão de memória semelhante ao computador pessoal e aos sistemas operativos UNIX, fornecendo ao mesmo tempo uma estrutura de integração em todas as informações armazenadas no servidor.

O sistema de ficheiros integrado compreende 11 sistemas de ficheiros, cada um com o seu próprio conjunto de estruturas e regras lógicas para interagir com informações armazenadas. Consulte Trabalhar com sistemas de ficheiros para obter mais informações.

As funções-chave do sistema de ficheiros integrado são:

- Suporte para armazenar informações em ficheiros de dados contínuos que podem conter cadeias de dados longas e contínuas. Estas cadeias de dados podem ser, por exemplo, o texto de um documento ou os elementos de imagem numa imagem. O suporte de ficheiros de dados contínuos está concebido de forma a permitir uma utilização eficiente nas aplicações de cliente/servidor.
- Uma estrutura de directórios hierárquica que permite uma organização dos objectos semelhante à disposição de frutos nos ramos de uma árvore. Ao especificar o caminho através dos directórios para um objecto, é permitido o acesso ao objecto.
- Uma interface comum que permite aos utilizadores e aplicações acederem não só aos ficheiros de dados contínuos, mas também a ficheiros de base de dados, documentos e outros objectos que estão armazenados no servidor.
- Uma vista comum dos ficheiros de dados contínuos que estão armazenados localmente no seu servidor, Integrated xSeries Server for iSeries ou num servidor de Windows NT<sup>®</sup> remoto. Os ficheiros de dados contínuos também podem ser armazenados remotamente num servidor de rede local (LAN), num servidor Novell NetWare, noutra servidor iSeries remoto ou num servidor de Sistemas de Ficheiros da Rede.

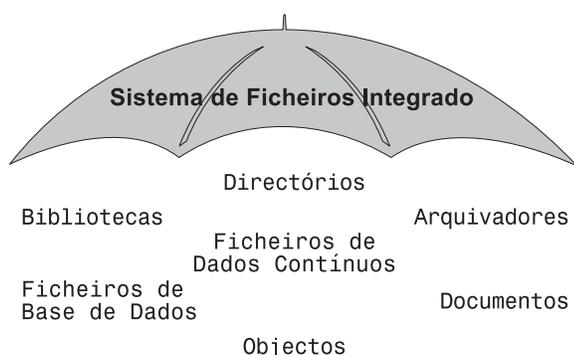


Figura 1. Uma estrutura de todas as informações armazenadas no servidor iSeries

## Porquê utilizar o sistema de ficheiros integrado?

O sistema de ficheiros integrado melhora as capacidades já extensas da gestão de dados do OS/400 com capacidades adicionais, de modo a melhorar o suporte das formas novas e futuras de processamento de informações, tais como cliente/servidor, sistemas abertos e multimedia.

Pode utilizar o sistema de ficheiros integrado para:

- Fornecer um acesso rápido a dados do OS/400, especialmente para aplicações como o Client Access, que utilizam o servidor de ficheiros do OS/400.
- Permitir um processamento mais eficiente de tipos de ficheiros de dados contínuos como, por exemplo, imagens, áudio e vídeo.
- Fornecer uma base de sistema de ficheiros e uma base de directórios para suporte de normas de sistema aberto baseado em UNIX, tais como POSIX (Portable Operating System Interface for Computer Environments) e XPG. Esta estrutura de ficheiros e esta estrutura de directórios também fornecem um ambiente familiar para os utilizadores de sistemas operativos de PC como, por exemplo, os sistemas operativos DOS (Disk Operating System) e Windows<sup>®</sup>.
- Permitir que ficheiros com capacidades únicas (tais como ficheiros de base de dados orientados para registos, ficheiros de dados contínuos com base em UNIX e servidor de ficheiros) sejam considerados como sistemas de ficheiros separados, sendo, no entanto, permitida a respectiva gestão através de uma interface comum.

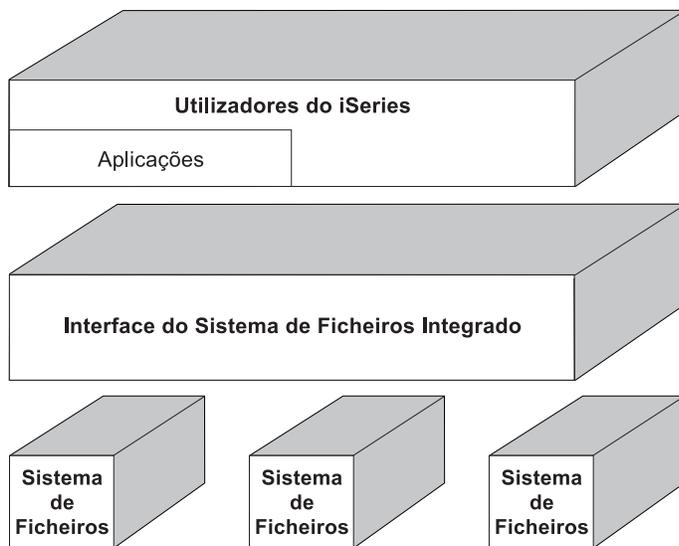


Figura 2. Interface comum para sistemas de ficheiros separados

- Permitir aos utilizadores de PC tirarem o melhor partido da interface gráfica de utilizador. Por exemplo, os utilizadores do Windows podem utilizar as ferramentas gráficas do Windows para trabalharem com ficheiros de dados contínuos do servidor iSeries e outros objectos da mesma forma que trabalham com os ficheiros armazenados nos PCs.
- Fornecer a continuidade dos nomes dos objectos e das informações associadas aos objectos em diversos idiomas nacionais. Por exemplo, isto garante que os caracteres individuais permaneçam inalterados quando se muda de uma página de códigos de um idioma para a página de códigos de outro idioma.

## Conceitos

Esta secção aborda conceitos do sistema de ficheiros integrado. Consulte qualquer um dos tópicos existentes na lista que se segue para obter mais informações sobre o funcionamento do sistema de ficheiros integrado no servidor.

- “Directório”
- “Ligação” na página 11
- “Nome de caminho” na página 15
- “Ficheiro de dados contínuos” na página 15
- “Continuidade de nomes” na página 17
- “Atributos expandidos” na página 17
- “Suporte de pesquisa” na página 18

## Directório

Um **directório** é um objecto especial que é utilizado para localizar objectos por nomes especificados pelo utilizador. Cada directório contém uma lista de objectos a ele ligados. Essa lista pode incluir outros directórios.

O sistema de ficheiros integrado fornece uma estrutura de directórios hierárquica que lhe permite aceder a todos os objectos do servidor. Podemos considerar que esta estrutura de directórios é uma árvore invertida, em que a raiz está para cima e os ramos para baixo. Os ramos representam directórios na hierarquia de directórios. Estes ramos de directórios têm ramos subordinados que se designam por subdirectórios. Ligados às várias ramificações de directórios e subdirectórios encontram-se objectos, tais como ficheiros. A localização de um objecto requer a especificação de um caminho através dos directórios

para o subdirectório ao qual o objecto está ligado. Os objectos que estão ligados a um determinado directório são, por vezes, descritos como estando “nesse” directório.

Um determinado ramo de directórios, respectivos ramos subordinados (subdirectórios) e todos os objectos a eles ligados, são designados por **subárvore**. Cada sistema de ficheiros é uma subárvore principal na estrutura de directórios do sistema de ficheiros integrado. Nas subárvores do sistema de ficheiros QSYS.LIB e QSYS.LIB de ASP independente, uma biblioteca é tratada da mesma forma que um subdirectório. Os objectos numa biblioteca são considerados como objectos num subdirectório. Uma vez que os ficheiros de base de dados contêm objectos (membros de ficheiros de base de dados), são considerados como subdirectórios e não como objectos. No sistema de ficheiros de serviços da biblioteca de documentos (subárvore QDLS), as pastas são tratadas como subdirectórios e os documentos em pastas são tratados como objectos num subdirectório.

Devido à existência de diferenças nos sistemas de ficheiros, as operações que podem ser executadas numa subárvore da hierarquia de directórios podem não funcionar noutra subárvore.

O suporte de directórios do sistema de ficheiros integrado é semelhante ao suporte de directórios fornecido pelo sistema de ficheiros do DOS. Adicionalmente, fornece funções específicas dos sistemas UNIX, tais como, a capacidade de armazenar um ficheiro uma vez só, mas aceder a esse ficheiro através de vários caminhos utilizando ligações.

Consulte os tópicos que se seguem para obter mais informações sobre os directórios do sistema de ficheiros integrado:

- “Directório actual” na página 7
- “Directório Inicial” na página 8
- “Directórios fornecidos” na página 8
- “Directórios \*TYPE2” na página 10

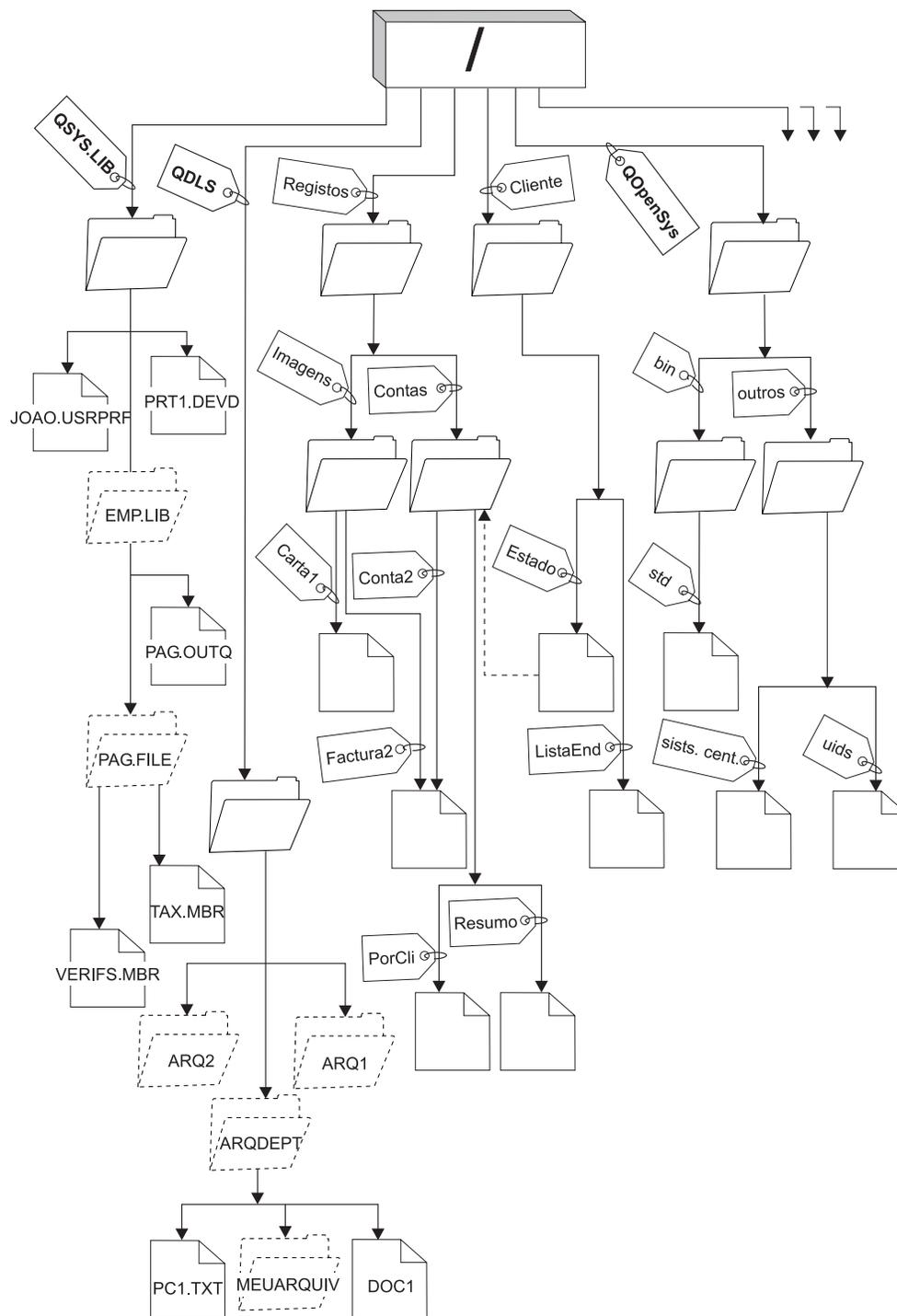


Figura 3. Os sistemas de ficheiros e objectos são ramos da árvore de directórios do sistema de ficheiros integrado

### Directório actual

O **directório actual** é o primeiro directório no qual o sistema operativo procura os programas e ficheiros, além de nele armazenar os ficheiros temporários e o output. Quando solicita uma operação para um objecto, como um ficheiro, o sistema procura o objecto no directório actual, a menos que especifique um caminho de directório diferente. O directório actual é semelhante à ideia de biblioteca actual. Também é chamado **directório de trabalho actual** ou apenas **directório de trabalho**.

## Directório Inicial

O **directório inicial** é utilizado como o directório actual quando inicia sessão no sistema. O nome do directório inicial é especificado no seu perfil do utilizador. Quando o trabalho é iniciado, o sistema procura no seu perfil de utilizador o nome do directório inicial. Se não existir um directório com esse nome no sistema, o directório inicial é alterado para o directório "raiz" (/).

Normalmente, o administrador do sistema que cria o perfil de utilizador para um utilizador também cria o directório inicial do utilizador. É recomendável criar directórios iniciais individuais para cada utilizador no directório /home. O directório /home é um subdirectório do directório "raiz" (/). O valor assumido do sistema assume que o nome do directório inicial de um utilizador é o mesmo que o perfil de utilizador.

Por exemplo, o comando `CRTUSRPRF USRPRF(Joao) HOMEDIR(*USRPRF)` atribuirá o directório inicial de João como /home/JOAO. Se o directório /home/JOÃO não existir, o directório "raiz" (/) torna-se o directório inicial do João.

Pode especificar, como directório actual, um directório diferente do directório inicial, em qualquer momento após iniciar sessão, utilizando o comando de CL Alterar Directório Actual (`CHGCURDIR`), a API `chdir()` ou a API `fchdir()`.

Como valor assumido, o directório inicial escolhido durante a iniciação do processo permanecerá o directório inicial de cada módulo. Isto independentemente de o seu perfil de utilizador activo para o módulo ter sido alterado após a iniciação. No entanto, é fornecido suporte pela API Alterar Trabalho (`QWTCHGJB`), que pode ser utilizada para alterar o directório inicial que está a ser utilizado por um módulo para o directório inicial do perfil do utilizador actual desse módulo (ou o directório "raiz" (/), se esse directório inicial não existir). Os módulos secundários herdarão sempre o directório inicial do módulo que o criou. Note que o directório actual do processo não é alterado quando utiliza `QWTCHGJB` para alterar o directório inicial do módulo. O directório actual tem como âmbito o nível do processo e o directório inicial tem como âmbito o nível do módulo. Alterar o directório de trabalho actual em qualquer módulo altera-o para todo o processo. Alterar o directório inicial de um módulo não altera o respectivo directório de trabalho actual.

Consulte o tópico Interfaces de programação de aplicações (APIs) para obter detalhes sobre a API `QWTCHGJB`.

## Directórios fornecidos

Quando o sistema é reiniciado, o sistema de ficheiros integrado cria os seguintes directórios, se ainda não existirem:

**/tmp** O directório /tmp fornece às aplicações um local onde armazenar os ficheiros temporários. Este é um subdirectório do directório "raiz" (/), de modo que o respectivo nome de caminho é /tmp.

Uma vez colocado por determinada aplicação no directório /tmp, o ficheiro permanece nesse directório até ser removido pelo utilizador ou por uma aplicação. O sistema não remove automaticamente ficheiros do /tmp nem executa nenhum outro processamento especial para os ficheiros no /tmp.

Pode utilizar os ecrãs e os comandos do utilizador que suportam o sistema de ficheiros integrado para gerir o directório /tmp e respectivos ficheiros. Por exemplo, pode utilizar o ecrã Trabalhar com Ligações de Objecto ou o comando `WRKLNK` para copiar, remover ou mudar o nome ao directório /tmp ou a ficheiros desse directório. É concedida a todos os utilizadores uma autoridade `*ALL` para o directório, o que significa que podem executar a maior parte das acções válidas no directório.

Uma aplicação pode utilizar as interfaces de programação de aplicações (APIs) que suportam o sistema de ficheiros integrado para gerir o /tmp e os respectivos ficheiros (consulte "Aceder utilizando APIs" na página 80). Por exemplo, o programa de aplicação pode remover um ficheiro do /tmp utilizando a API `desligar()`.

Se o /tmp for removido, será automaticamente recriado durante o próximo reinício do sistema.

O directório /tmp é agora enviado juntamente com o atributo "restringir a mudança de nome e desligar" activado. O comando Alterar Atributo, CHGATR, pode ser utilizado para alterar este atributo. Os utilizadores têm permissão para desactivar este atributo, mas é automaticamente reactivado durante o próximo reinício do sistema. Quando este atributo está activado, o directório /tmp tem a seguinte restrição:

- As mudanças de nome e remoções de ligações estão restritas para objectos deste directório. É possível ligar objectos neste directório, mas não é possível mudar o nome ou remover as respectivas ligações a menos que uma ou mais das seguintes condições seja verdadeira de forma a permitir ao utilizador executar a operação:

1. O utilizador é o proprietário do objecto
2. O utilizador é o proprietário do directório
3. O utilizador tem uma autoridade especial \*ALLOBJ

**/home** Os administradores do sistema utilizam o directório /home para armazenar um directório diferente para cada utilizador. Normalmente, o administrador do sistema define o directório inicial que está associado ao perfil de utilizador como o directório do utilizador no /home, por exemplo/home/joao. Consulte "Directório Inicial" na página 8 para obter mais informações.

**/etc** O directório /etc armazena ficheiros administrativos, de configuração e outros ficheiros de sistema.

**/usr** O directório /usr inclui subdirectórios que contêm informações utilizadas pelo sistema. Normalmente, os ficheiros no /usr não são alterados frequentemente.

**/usr/bin**  
O directório /usr/bin contém os programas utilitários padrão.

**/QIBM**  
O directório /QIBM é o directório do sistema e é fornecido com o sistema.

**/QIBM/ProdData**  
O directório /QIBM/ProdData é um directório do sistema utilizado para dados do Programa Licenciado.

**/QIBM/UserData**  
O directório /QIBM/UserData é um directório do sistema utilizado para dados do utilizador do Programa Licenciado como, por exemplo, ficheiros de configuração.

**/QOpenSys/QIBM**  
O directório /QOpenSys/QIBM é o directório de sistema para o sistema de ficheiros QOpenSys.

**/QOpenSys/QIBM/ProdData**  
O directório /QOpenSys/QIBM/ProdData é o directório do sistema para o sistema de ficheiros QOpenSys e é utilizado para dados do Programa Licenciado.

**/QOpenSys/QIBM/UserData**  
O directório /QOpenSys/QIBM/UserData é o directório do sistema para o sistema de ficheiros QOpenSys e é utilizado para dados do utilizador do Programa Licenciado como, por exemplo, ficheiros de configuração.

**/nome\_esp/QIBM**  
O directório /nome\_esp/QIBM é o directório do sistema para quaisquer ASPs independentes existentes no seu sistema, em que nome\_esp é o nome do ASP independente.

**/nome\_esp/QIBM/UserData**  
O directório /nome\_esp/QIBM/UserData é um directório de sistema utilizado para dados do utilizador do Programa Licenciado como, por exemplo, ficheiros de configuração para quaisquer ASPs independentes existentes no seu sistema, em que nome\_esp é o nome do ASP independente.

**/dev** O directório /dev contém vários ficheiros e directórios de sistema.

## **/dev/xti**

O directório /dev/xti contém os controladores de dispositivos UDP e TCP. Consulte "Dispositivos UDP e TCP no sistema de ficheiros "raiz" (/)" na página 30 para obter mais informações.

## **Directórios \*TYPE2**

Os sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros definidos pelo utilizador (UDFS) do sistema de ficheiros integrado suportam o formato de directório \*TYPE2. O formato de directório \*TYPE2 é um melhoramento do formato de directório \*TYPE1 original. Os directórios \*TYPE2 têm uma estrutura interna e uma implementação diferentes dos directórios \*TYPE1.

As vantagens dos directórios \*TYPE2 são:

- Aumento do rendimento
- Aumento da fiabilidade
- Funcionalidade acrescida
- Menos espaço de memória auxiliar (em muitos casos).

Os directórios \*TYPE2 melhoram o rendimento do sistema de ficheiros relativamente aos directórios \*TYPE1, especialmente na criação e eliminação de directórios.

Os directórios \*TYPE2 são mais fiáveis que os directórios \*TYPE1. Após o fim anormal de um sistema, os directórios \*TYPE2 são totalmente recuperados, a menos que tenha ocorrido uma falha na memória auxiliar. Os directórios \*TYPE1 podem requerer a utilização do comando Regenerar Memória (RCLSTG) para poderem recuperar totalmente.

Os directórios \*TYPE2 fornecem a seguinte funcionalidade acrescida:

1. Os directórios \*TYPE2 suportam a mudança de tipo de letra de um nome num sistema de ficheiros de tipo de letra único (por exemplo, a mudança de A para a).
2. Um objecto num directório \*TYPE2 pode ter até um milhão de ligações, em comparação com as 32.767 ligações possíveis nos directórios \*TYPE1. Isto significa que pode ter até 1 milhão de ligações permanentes a um ficheiro de dados contínuos e um directório \*TYPE2 pode conter até 999.998 subdirectórios.
3. Através do iSeries Navigator, quando abre um directório com o formato \*TYPE2, a lista de entradas é automaticamente ordenada em sequência binária.
4. Algumas funções novas, tais como o suporte de pesquisa do sistema de ficheiros integrado, só estão disponíveis para os objectos existentes nos directórios \*TYPE2.

Normalmente, os directórios \*TYPE2 com menos de 350 objectos requerem menos memória auxiliar do que os directórios \*TYPE1 com o mesmo número de objectos. Os directórios \*TYPE2 com mais de 350 objectos são dez por cento maiores (em média) do que os directórios \*TYPE1.

Existem várias formas de obter directórios \*TYPE2 no seu sistema:

- Os novos servidores iSeries que sejam pré-instalados com o OS/400 V5R2 ou posterior têm directórios \*TYPE2. Não é necessária a conversão do directório "raiz" (/), do QOpenSys e dos UDFSs nos ASPs 1-32.
- Uma instalação de raiz do OS/400 V5R2 ou posterior num iSeries server tem directórios \*TYPE2. Não é necessária a conversão do directório "raiz" (/), do QOpenSys e dos UDFSs nos ASPs 1-32.
- Foi utilizado o utilitário de conversão da V5R1 ou V5R2 para converter os sistemas de ficheiros.
- Se os UDFSs existentes num conjunto de memória auxiliar (ASP) independente ainda não tiverem sido convertidos para o formato \*TYPE2, sê-lo-ão na primeira vez que o ASP independente for activado num sistema com o OS/400 V5R2, ou posterior, instalado.
- Todos os outros sistemas de ficheiros suportados, excepto os UDFSs em ASPs independentes que ainda estejam a utilizar directórios \*TYPE1, serão convertidos automaticamente pelo sistema. Esta conversão

terá início após a instalação da V5R3 e não deverá afectar as actividades do sistema. Consulte “Converter directórios de \*TYPE1 para \*TYPE2” na página 84 para obter mais informações.

Para determinar o formato de directório dos sistemas de ficheiros existentes no seu servidor, utilize o comando Converter Directório (CVTDIR):

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

**Nota:** Os directórios \*TYPE2 são suportados no OS/400 V5R1 ou posterior, mas existem algumas diferenças em relação ao suporte de directório \*TYPE2 normal. Para obter mais informações, consulte Utilizar directórios \*TYPE2 no OS/400 V5R1 ou V5R2.

Para obter mais informações sobre directórios \*TYPE2, consulte “Converter directórios de \*TYPE1 para \*TYPE2” na página 84.

**Utilizar directórios \*TYPE2 no OS/400 V5R1 ou V5R2:** Os sistemas de ficheiros “raiz” (/), QOpenSys e os sistemas de ficheiros definidos pelo utilizador (UDFS) no sistema de ficheiros integrado suportam o formato de directório \*TYPE2 no OS/400 V5R1, V5R2 e posterior. O formato de directório \*TYPE2 é um melhoramento do formato de directório \*TYPE1 original. Os directórios \*TYPE2 têm uma estrutura interna diferente dos directórios \*TYPE1 e facultam um aumento do rendimento e fiabilidade.

Se tiver a V5R1 ou V5R2, poderá converter os directórios para o formato de directório \*TYPE2 através do utilitário de conversão adequado. Pouco tempo depois de instalada a V5R3, é iniciada automaticamente a conversão para directórios \*TYPE2 de qualquer um dos sistemas de ficheiros que ainda não tenham sido convertidos para suportarem directórios \*TYPE2. Assim, convém converter para o formato de directório \*TYPE2 antes de instalar a V5R3, de forma a evitar esta conversão automática. Para obter mais informações, consulte “Converter directórios de \*TYPE1 para \*TYPE2” na página 84.

O suporte de directórios \*TYPE2 na V5R2 está disponível no Centro de Informações do iSeries V5R2 através do comando Converter directório (CVTDIR).

O suporte de directórios \*TYPE2 na V5R1 está disponível através de correcções (PTFs). O utilitário de conversão é ligeiramente diferente do da versão V5R2. Consulte o APAR informativo II13161 para obter a documentação completa sobre os directórios \*TYPE2 na V5R1. Utilize um dos seguintes métodos para ter acesso ao APAR:

1. Descarregue o APAR informativo para o iSeries server e consulte-o. Utilize os seguintes comandos:

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

2. Consulte o sítio da Web

<http://www.ibm.com/eserver/iseries/support/supporthome.nsf/document/10000045>  para ver o APAR informativo online. Consulte **Problem Solving** —> **Technical Databases** —> **Authorized Program Analysis Reports (APARs)** —> **V5R1 APARs** —> **APAR number II13161**.

## Ligação

Uma **ligação** é uma ligação com nome entre um directório e um objecto. Um utilizador ou programa pode indicar ao servidor onde encontrar um objecto, especificando o nome de uma ligação ao objecto. Uma ligação pode ser utilizada como um nome de caminho ou parte de um nome de caminho.

Para os utilizadores dos sistemas de ficheiros baseados em directórios, é conveniente pensar num objecto, por exemplo, um ficheiro, como algo que tem um nome que o identifica perante o servidor. De facto, é o caminho do directório para o objecto que o identifica. Por vezes, pode aceder a um objecto indicando apenas o “nome” do objecto. Pode fazer isto apenas porque o sistema está concebido de forma a assumir

a parte do directório do caminho, em determinadas condições. A ideia de uma ligação tira partido da realidade de que é o caminho do directório que identifica o objecto. O nome é atribuído à ligação e não ao objecto.

Uma vez familiarizado com a ideia de que é à ligação que é atribuído o nome e não ao objecto, começa a ver possibilidades que não eram tão evidentes. Podem existir múltiplas ligações ao mesmo objecto. Por exemplo, dois utilizadores podem partilhar um ficheiro, tendo uma ligação ao mesmo a partir do directório inicial de cada utilizador (consulte “Directório Inicial” na página 8). Determinados tipos de ligações podem cruzar sistemas de ficheiros e podem existir sem haver um objecto.

Existem dois tipos de ligações: “Ligação permanente” e “Ligação simbólica” na página 13. Quando utilizar nomes de caminho em programas, pode optar por utilizar uma ligação permanente ou uma ligação simbólica. Cada tipo de ligação tem vantagens e desvantagens. As condições sob as quais um tipo de ligação tem uma vantagem sobre o outro são as seguintes:

*Tabela 1. Comparação entre Ligação Permanente e Ligação Simbólica*

Item	Ligação Permanente	Ligação Simbólica
Resolução do nome	Mais rápida. Uma ligação permanente contém uma referência directa ao objecto.	Mais lenta. Uma ligação simbólica contém um nome de caminho para o objecto, que tem de ser resolvido para encontrar o objecto.
Existência do objecto	Necessária. Tem de existir um objecto para criar uma ligação permanente ao mesmo.	Opcional. Uma ligação simbólica pode ser criada quando o objecto ao qual se refere não existe.
Eliminação do objecto	Limitada. Todas as ligações permanentes têm de ser desligadas (removidas) para eliminar o objecto.	Ilimitada. É possível eliminar um objecto mesmo que haja ligações simbólicas a ele referentes.
Objectos estáticos (os atributos não são alterados)	Mais rápida. Para um objecto estático, a resolução do nome é a prioridade da execução principal. A resolução do nome é mais rápida quando são utilizadas ligações permanentes.	Mais lenta. A resolução do nome é mais lenta quando são utilizadas ligações simbólicas.
Âmbito	Limitada. As ligações permanentes não podem cruzar sistemas de ficheiros.	Ilimitada. As ligações simbólicas podem cruzar sistemas de ficheiros.

Consulte os tópicos que se seguem para obter mais informações sobre ligações:

- Ligação permanente
- Ligação simbólica

## Ligação permanente

Uma **ligação permanente**, que é, por vezes, chamada apenas ligação, não poderá existir se não for ligada a um objecto real. Quando um objecto é criado num directório (por exemplo, copiando um ficheiro para um directório), é estabelecida a primeira ligação permanente entre o directório e o objecto. Os utilizadores e as aplicações podem adicionar outras ligações permanentes. Cada ligação permanente é indicada por uma entrada de directório diferente no directório. As ligações do mesmo directório não podem ter o mesmo nome, mas as ligações de diferentes directórios podem ter o mesmo nome.

Caso o sistema de ficheiros o suporte, podem existir múltiplas ligações permanentes a um objecto, a partir do mesmo directório ou a partir de diferentes directórios. A única excepção ocorre quando o objecto existe noutra directório. Pode haver apenas uma ligação permanente de um directório para outro directório.

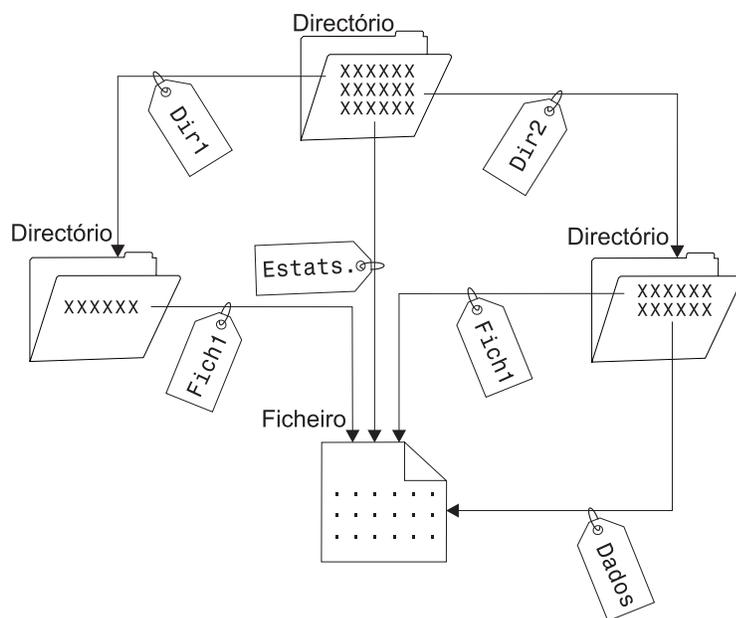


Figura 4. Uma entrada de directório define cada ligação permanente.

As ligações permanentes podem ser removidas sem afectar a existência de um objecto, desde que exista, no mínimo, uma ligação permanente ao objecto. Quando a última ligação permanente é removida, o objecto é removido do servidor, a não ser que uma aplicação mantenha o objecto aberto. Cada aplicação que mantenha o objecto aberto pode continuar a utilizá-lo até que essa aplicação feche o objecto. Quando o objecto é fechado pela última aplicação que o utiliza, é removido do servidor. Um objecto não pode ser aberto depois de a última ligação permanente ter sido removida.

O conceito de ligação permanente também pode ser aplicado aos sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente e ao sistema de ficheiros de serviços da biblioteca de documentos (QDLS), mas com uma restrição. De facto, uma biblioteca tem uma ligação permanente a cada objecto na biblioteca. Da mesma forma, uma pasta tem uma ligação permanente a cada documento existente na pasta. No entanto, não são permitidas múltiplas ligações permanentes ao *mesmo objecto* em QSYS.LIB, QSYS.LIB de ASP independente ou no QDLS.

Uma ligação permanente não pode cruzar sistemas de ficheiros. Por exemplo, um directório no sistema de ficheiros QOpenSys não pode ter uma ligação permanente a um objecto nos sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente ou a um documento no sistema de ficheiros QDLS.

### Ligação simbólica

Uma ligação simbólica, também conhecida como ligação não permanente, é um nome de caminho contido num ficheiro. Quando o sistema encontra uma ligação simbólica, segue o nome de caminho fornecido pela ligação simbólica e, em seguida, continua em qualquer caminho restante que segue a ligação simbólica. Se o nome de caminho começar por /, o sistema volta ao directório / ("raiz") e começa a seguir o caminho a partir desse ponto. Se o nome de caminho não começar por /, o sistema regressa ao directório imediatamente anterior e segue o nome de caminho na ligação simbólica que começa nesse directório.

Considere o exemplo seguinte relativo à utilização de uma ligação simbólica:

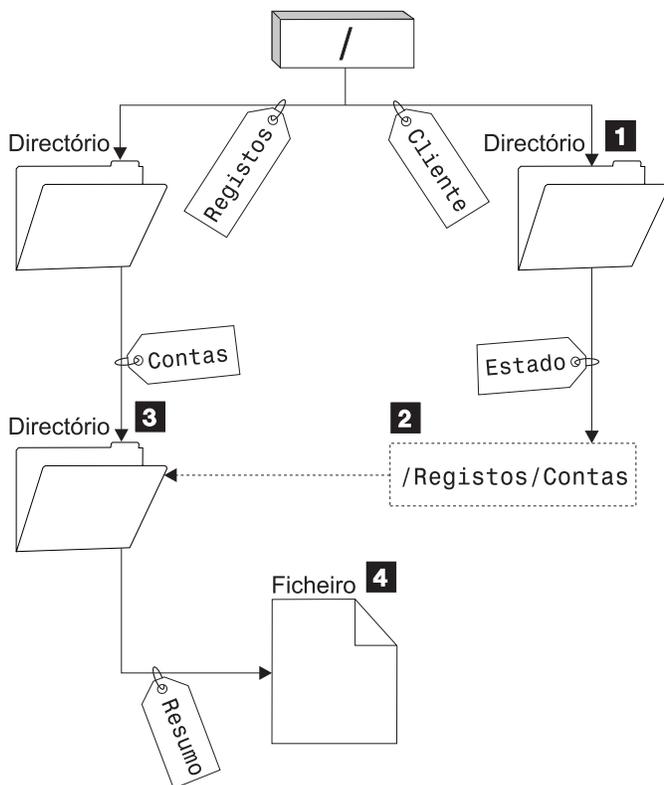


Figura 5. Exemplo de utilização de uma ligação simbólica

Pode seleccionar uma opção de menu para verificar o estado das contas dos clientes. O programa que apresenta o menu utiliza o nome de caminho seguinte:

`/Cliente/Estado/Resumo`

O sistema segue a ligação *Cliente*, que conduz a um directório **1** e, em seguida, segue a ligação *Estado*. A ligação *Estado* é uma ligação simbólica, que contém um nome de caminho **2**. Uma vez que o nome de caminho começa por /, o sistema regressa ao directório / ("raiz") e segue as ligações *Registos* e *Contas* sequencialmente. Este caminho conduz a outro directório **3**. Neste ponto, o sistema conclui o caminho no nome de caminho fornecido pelo programa. Segue a ligação *Resumo*, que conduz a um ficheiro **4** contendo os dados que são necessários.

Ao contrário de uma ligação permanente, uma ligação simbólica é um objecto (do tipo de objecto \*SYMLNK); pode existir sem apontar para um objecto que exista. Pode utilizar uma ligação simbólica, por exemplo, para fornecer um caminho para um ficheiro que será adicionado ou substituído mais tarde.

Ao contrário de uma ligação permanente, uma ligação simbólica pode cruzar sistemas de ficheiros. Por exemplo, se estiver a trabalhar num sistema de ficheiros, pode utilizar uma ligação simbólica para ter acesso a um ficheiro noutro sistema de ficheiros. Embora os sistemas de ficheiros QSYS.LIB, QSYS.LIB de ASP independente e QDLS não suportem a criação e armazenamento de ligações simbólicas, poderá criar uma ligação simbólica no sistema de ficheiros "raiz" (/) ou QOpenSys que lhe permita:

- Aceder a um membro de ficheiro de base de dados nos sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente.
- Aceder a um documento no sistema de ficheiros QDLS.

Consulte também a Tabela 1 na página 12.

## Nome de caminho

Um **nome de caminho** indica ao servidor como localizar um objecto. O nome de caminho é expresso como uma sequência de nomes de directórios, seguida do nome do objecto. Os directórios individuais e o nome do objecto são separados por uma barra (/); por exemplo:

```
directório1/directório2/ficheiro
```

Por uma questão de conveniência, é possível utilizar a barra invertida (\) em vez da barra, nos comandos do sistema de ficheiros integrado.

Há duas formas de indicar um nome de caminho:

- Um **nome de caminho absoluto** começa no nível mais elevado ou o directório “raiz” (que é identificado pelo carácter /). Por exemplo, considere o caminho seguinte desde o directório / até ao ficheiro com o nome Silva.

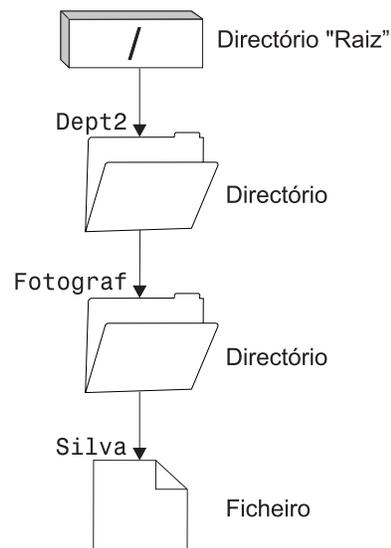


Figura 6. Componentes de um nome de caminho

O nome de caminho absoluto para o ficheiro Silva é o seguinte:

```
/Dept2/Fotograf/Silva
```

O nome de caminho absoluto é também conhecido por **nome de caminho completo**.

- Se o nome de caminho não começar pelo carácter /, o sistema assume que o caminho começa no directório actual. Este tipo de nome de caminho é denominado **nome de caminho relativo**. Por exemplo, se o directório actual for Dept2 e tiver um subdirectório Fotograf com o ficheiro Silva, o nome de caminho relativo para o ficheiro é:

```
Fotograf/Silva
```

Repare que o nome de caminho não inclui o nome do directório actual. O primeiro item do nome é o directório ou objecto no *nível seguinte abaixo* do directório actual.

## Ficheiro de dados contínuos

Um **ficheiro de dados contínuos** é uma sequência de bytes acessível aleatoriamente, sem qualquer estrutura adicional imposta pelo sistema. O sistema de ficheiros integrado fornece suporte para armazenar e trabalhar com informações sob a forma de ficheiros de dados contínuos. Os documentos armazenados nas pastas do servidor são ficheiros de dados contínuos. Outros exemplos de ficheiros de dados contínuos são os ficheiros de PC e os ficheiros dos sistemas UNIX. Um ficheiro de dados contínuos do sistema de ficheiros integrado é um objecto do sistema que tem um tipo de objecto \*STMF.

Para compreender melhor os ficheiros de dados contínuos, será útil compará-los com os ficheiros de base de dados do iSeries. Um ficheiro de base de dados é orientado para os registos, tem subdivisões predefinidas constituídas por um ou mais campos com características específicas, tais como o comprimento e o tipo de dados.



Figura 7. Comparação de um ficheiro de dados contínuos com um ficheiro orientado para registos

Os ficheiros de dados contínuos e os ficheiros orientados para registos são estruturados de formas diferentes e esta diferença na estrutura afecta o modo como os ficheiros são utilizados. A estrutura afecta a forma como uma aplicação é escrita para interagir com os ficheiros, bem como a melhor localização para a utilização de cada tipo de ficheiro numa aplicação. Um ficheiro orientado para registos, por exemplo, é mais adequado para armazenamento de estatísticas de clientes, tais como o nome, endereço e saldo de conta. Um ficheiro orientado para registos permite que estes campos predefinidos sejam acedidos e manipulados individualmente, utilizando as funções de programação expandidas do servidor. Mas um ficheiro de dados contínuos é mais adequado para armazenamento de informações, tais como uma imagem do cliente, compostas por uma cadeia contínua de bits representando variações de cor. Os ficheiros de dados contínuos são particularmente adequados para armazenamento de cadeias de dados, tais como o texto de um documento, imagens, áudio e vídeo.

Um ficheiro tem uma de duas opções de formato: ficheiro de dados contínuos \*TYPE1 ou \*TYPE2. O formato de ficheiro depende da edição em que foi criado o ficheiro ou, se um ficheiro foi criado num sistema de ficheiros definido pelo utilizador, do valor que foi especificado nesse sistema de ficheiros.

Para obter mais informações sobre ficheiros de dados contínuos no sistema de ficheiros integrado, consulte:

- “Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados” na página 94.
- “Ficheiros de dados contínuos \*TYPE1”.
- “Ficheiros de dados contínuos \*TYPE2”.

### Ficheiros de dados contínuos \*TYPE1

Um ficheiro de dados contínuos \*TYPE1 tem o mesmo formato que os ficheiros de dados contínuos criados em edições anteriores à versão 4, edição 4 do OS/400. Tem um tamanho mínimo de 4096 bytes. Os ficheiros de dados contínuos \*TYPE1 têm um tamanho de objectos máximo de, aproximadamente, 256 gigabytes.

### Ficheiros de dados contínuos \*TYPE2

Um ficheiro de dados contínuos \*TYPE2 tem acesso a ficheiros de alto rendimento e era novo na versão 4, edição 4, do OS/400. Na V5R3, os ficheiros de dados contínuos \*TYPE2 têm agora um tamanho de

objectos máximo de aproximadamente 1 terabyte nos sistemas de ficheiros "raiz" (/), QOpenSys e nos sistemas de ficheiros definidos pelo utilizador. De outro modo, o tamanho máximo é de aproximadamente 256 gigabytes. Tem igualmente capacidade de definição da memória, bem como de especificação de um atributo para otimizar a atribuição de memória principal. Todos os ficheiros criados em sistemas com a V4R4 e sistemas mais recentes são ficheiros de dados contínuos \*TYPE2, excepto se foram criados num sistema de ficheiros definido pelo utilizador, no qual era especificado um formato de ficheiro \*TYPE1.

**Nota:** Os ficheiros com um tamanho acima dos 256 gigabytes não podem ser guardados nem restaurados em sistemas com versões anteriores à V5R3.

## Continuidade de nomes

Quando utiliza os sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que garante que os caracteres nos nomes de objectos permanecem iguais. Isto também se aplica à utilização destes sistemas de ficheiros no iSeries server e nos dispositivos ligados que têm diferentes esquemas de codificação de caracteres (páginas de códigos). O seu servidor armazena os caracteres dos nomes num formato de 16 bits que é conhecido como UCS2 Nível 1 (também denominado **Unicode**) para directórios \*TYPE1 e UTF-16 para directórios \*TYPE2. Consulte Directórios \*TYPE2 para obter mais informações sobre os formatos de directório. O UCS2 Nível 1 e o UTF-16 são subconjuntos da norma ISO 10646. Quando o nome é utilizado, o sistema converte o formato armazenado dos caracteres na representação dos caracteres adequada na página de códigos a ser utilizada. Os nomes de atributos expandidos associados a cada objecto também são tratados do mesmo modo.

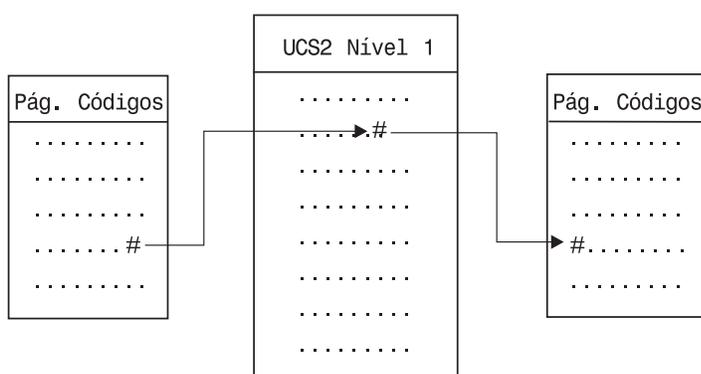


Figura 8. Manter caracteres inalterados independentemente dos esquemas de codificação

Este suporte facilita a interacção com um servidor a partir de dispositivos que utilizam diferentes páginas de códigos. Por exemplo, os utilizadores de PC podem ter acesso a um ficheiro do servidor iSeries utilizando o mesmo nome de ficheiro, mesmo que os respectivos PCs não tenham a mesma página de códigos que o servidor. A conversão de uma página de códigos noutra é executada automaticamente pelo seu servidor. É óbvio que o dispositivo tem de utilizar uma página de códigos que contenha os caracteres utilizados no nome.

## Atributos expandidos

Um atributo expandido (EA) corresponde a informações associadas a um objecto que fornecem detalhes adicionais sobre o objecto. O EA consiste num nome, que é utilizado para se referir a ele, e num valor. O valor pode ser texto, dados binários ou outro tipo de dados.

Os EAs de um objecto existem apenas enquanto o objecto existir.

Os EAs aparecem sob diversas formas e podem ser utilizados para conter várias informações. Poderá ser necessário conhecer os três EAs seguintes, nomeadamente:

## **.SUBJECT**

Uma breve descrição do conteúdo ou da finalidade do objecto.

**.TYPE** O tipo de dados no objecto. O tipo de dados pode ser texto, binário, código fonte para um programa, um programa compilado ou outras informações.

## **.CODEPAGE**

A página de códigos a ser utilizada para o objecto. A página de códigos utilizada para o objecto também é utilizada para os EAs associados ao objecto.

Um ponto (.) como primeiro carácter do nome significa que o EA é um EA padrão do sistema (SEA), que está reservado para utilização pelo sistema.

Os vários objectos nos vários sistemas de ficheiros podem ou não ter EAs. Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente suportam três EAs predefinidos: .SUBJECT, .TYPE e .CODEPAGE. No sistema de ficheiros de serviços da biblioteca de documentos (QDLS), as pastas e documentos podem ter qualquer tipo de EA. Algumas pastas e documentos poderão ter EAs, outros não. Nos sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, todos os directórios, ficheiros de dados contínuos e ligações simbólicas podem ter qualquer tipo de EAs. No entanto, alguns poderão nem sequer ter EAs.

O comando Trabalhar com Ligações de Objecto (WRKLNK) pode ser utilizado para visualizar o atributo expandido (EA) .SUBJECT de um objecto. Não existe qualquer outro suporte do sistema de ficheiros integrado através do qual as aplicações ou os utilizadores possam aceder e alterar os EAs. As únicas excepções a esta regra são os comandos de CL Ver UDFS (DSPUDFS) e Ver Informações sobre Sistemas de Ficheiros Instalados (DSPMFSINF), que apresentam os atributos expandidos aos utilizadores.

No entanto, os EAs associados a alguns objectos no QDLS podem ser alterados através de interfaces fornecidas pelo sistema de ficheiros hierárquico (HFS). Consulte “Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)” na página 43 e “Sistema de Ficheiros Óptico (QOPT)” na página 45 para obter mais informações sobre estes sistemas de ficheiros.

Se um PC cliente estiver ligado a um servidor iSeries através do OS/2® ou do Windows, é possível utilizar as interfaces de programação do respectivo sistema operativo (por exemplo, DosQueryFileInfo e DosSetFileInfo) para consultar e definir os EAs de qualquer objecto de ficheiro. Os utilizadores do OS/2 podem também alterar os EAs de um objecto no ambiente de trabalho, utilizando o bloco de notas de definições; isto é, seleccionando Definições no menu de sobreposição associado ao objecto.

Se definir atributos expandidos, utilize as seguintes directrizes de nomenclatura:

- O nome de um EA pode ter um máximo de 255 caracteres de comprimento.
- Não utilize um ponto (.) como primeiro carácter do nome. Um EA cujo nome comece por um ponto é interpretado como sendo um EA padrão do sistema.
- Para minimizar a possibilidade de conflitos de nomes, utilize uma estrutura de nomenclatura coerente para os EAs. Recomenda-se o seguinte formato:

NomeEmpresaNomeProduto.Nome\_Atributo

## **Suporte de pesquisa**

Agora o iSeries dispõe da capacidade de permitir a pesquisa de objectos do sistema de ficheiros integrado. Este novo suporte proporciona maior flexibilidade aos utilizadores do iSeries: permite a execução de pesquisas de vários itens; os utilizadores é que decidem quando pretendem que seja efectuada a pesquisa; os utilizadores é que decidem que acções efectuar, com base nos resultados da pesquisa. Para obter mais informações sobre pesquisas, consulte os seguintes tópicos:

- “Valores do sistema relacionados” na página 20
- “Ocorrências de pesquisa” na página 21
- “Exemplos de pesquisa” na página 19

Os dois novos pontos de saída relacionados com este suporte são:

- QIBM\_QP0L\_SCAN\_OPEN - Pesquisa do Sistema de Ficheiros Integrado num Programa de Saída Aberto

Para este ponto de saída, é utilizada a pesquisa do sistema de ficheiros integrado em programas de saída abertos, para efectuar o processamento de pesquisa quando um objecto do sistema de ficheiros integrado é aberto em determinadas condições.

- QIBM\_QP0L\_SCAN\_CLOSE - Pesquisa do Sistema de Ficheiros Integrado num Programa de Saída Fechado

Para este ponto de saída, é utilizada a pesquisa do sistema de ficheiros integrado em programas de saída fechados, para efectuar o processamento de pesquisa quando um objecto do sistema de ficheiros integrado é fechado em determinadas condições.

Para obter mais informações sobre como definir se os objectos deverão ou não ser pesquisados, consulte "Definir se os objectos devem ou não ser pesquisados" na página 119.

**Nota:** Apenas serão pesquisados os objectos dos sistemas de ficheiros que tenham sido totalmente convertidos para directórios \*TYPE2.

## Exemplos de pesquisa

Seguem-se dois exemplos de pesquisas do programa de saída:

- Vírus

Os programas de saída podem verificar a existência de vírus. Se for localizado um vírus num ficheiro, o programa anti-vírus pode reparar o problema ou tentar colocar o vírus em quarentena. Uma vez que o iSeries não poderá ser contaminado pelo vírus, este procedimento tem como objectivo reduzir as transmissões de vírus entre os servidores.

- Chamadas para saber quando foi aberto um ficheiro

Também poderá efectuar uma pesquisa para saber quando foi aberto um ficheiro. Ao efectuar esta pesquisa, é possível controlar a data e a hora de acesso a determinados ficheiros. Este procedimento será útil se pretender controlar o comportamento de determinados utilizadores.

A pesquisa poderá ocorrer em dois períodos de tempo diferentes, consoante a definição dos valores do sistema e a maneira como foi estabelecido o ambiente de pesquisa. A seguir é apresentada uma descrição dos dois períodos de tempo diferentes.

### 1. Pesquisa em tempo de execução

Uma pesquisa em tempo de execução é uma pesquisa de um ou mais ficheiros durante as actividades diárias normais. Desta forma, assegurará a integridade dos ficheiros sempre que sejam acedidos. A pesquisa durante as actividades normais garante que o ficheiro ou ficheiros se encontram actualizados independentemente dos padrões pesquisados.

#### Exemplo de verificação da existência de vírus em tempo de execução

Numa quarta-feira, acede a um ficheiro do sistema de ficheiros integrado a partir do PC. Quando abre o ficheiro a partir do PC, o ficheiro é pesquisado, uma vez que está registado um programa de saída aberto e o valor do sistema QSCANFS está definido para pesquisar ficheiros nos sistemas de ficheiros "raiz" (/), QOpenSys e UDFS. A pesquisa encontrou um vírus e o programa de saída anti-vírus procedeu à reparação do problema. Uma vez que o programa de saída reparou o ficheiro e este já não se encontra contaminado, o acesso a partir do PC não será contaminado nem poderá propagar a contaminação.

E, se em vez de verificar a existência de vírus nesse acesso, optasse por não efectuar uma pesquisa em tempo de execução? Depois de aceder ao ficheiro contaminado a partir do PC, o vírus foi transferido para o PC. Ao efectuar uma pesquisa em tempo de execução, o vírus é detectado antes de se propagar pelo PC.

O principal problema deste método é a necessidade de tempo de recurso para efectuar as pesquisas. Quem tentar aceder a um ficheiro terá de aguardar até à conclusão da pesquisa, antes de utilizar o

ficheiro. O sistema tenta garantir que a pesquisa seja efectuada apenas quando necessário e não em todos os acessos. Consulte QIBM\_QP0L\_SCAN\_OPEN e QIBM\_QP0L\_SCAN\_CLOSE para obter mais informações.

## 2. Pesquisa activada manualmente ou em massa

Deverá utilizar esta opção se pretende pesquisar vários itens de uma só vez. Neste caso, pode definir a execução da pesquisa para o fim de semana, quando o servidor não estiver em funcionamento. Deste modo, não terá qualquer problema em aceder aos ficheiros durante as actividades diárias normais. Uma vez que a pesquisa é efectuada offline, este procedimento poderá reduzir o tempo da pesquisa em tempo de execução em relação a ficheiros que não sejam alterados após a pesquisa em massa, dado que não é requerida uma nova pesquisa quando os ficheiros são acedidos.

## Valores do sistema relacionados

Existem dois novos valores de sistema relacionados com este novo suporte de pesquisa. Poderá utilizar estes dois valores de sistema para definir o ambiente de pesquisa que pretende para o servidor. Abaixo encontra-se uma lista de nomes dos dois valores de sistema e as respectivas descrições. Poderá visualizar uma descrição destes valores de sistema e das respectivas opções de controlo para o iSeries Navigator. Os valores de interface com base nos caracteres comparáveis são listados entre parênteses a seguir aos nomes do iSeries Navigator. Por exemplo, para o valor de sistema QSCANFSCTL, quando a opção de controlo do iSeries Navigator 'Pesquisar acessos apenas através dos servidores de ficheiros' é seleccionada origina, na prática os mesmos resultados, que a especificação de \*FSVROONLY, a opção de controlo com base nos caracteres.

O nome e as respectivas descrições são apresentados a seguir:

### 1. Utilize o programa de saída registado para pesquisar os sistemas de ficheiros "raiz" (/), QOpenSys e os sistemas de ficheiros definidos pelo utilizador (QSCANFS)

Este valor de sistema pode ser utilizado para especificar quando devem ser pesquisados os sistemas de ficheiros. Só serão pesquisados os objectos dos sistemas de ficheiros "raiz" (/), QOpenSys e dos sistemas de ficheiros definidos pelo utilizador se o sistema de ficheiros já tiver sido totalmente convertido. Este valor especifica quando devem ser pesquisados os objectos pelos programas de saída registados com qualquer um dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado.

Como valor assumido, os objectos serão pesquisados se estiverem registados programas de saída.

### 2. Controlo de pesquisa (QSCANFSCTL)

Para este valor de sistema, pode utilizar as opções de controlo assumidas ou as opções de controlo especificadas. Para obter descrições abreviadas das diferentes opções de controlo especificadas com base nos valores de sistema do iSeries Navigator, consulte abaixo.

- Pesquisar acessos apenas através dos servidores de ficheiros - (\*FSVROONLY especificado)

Só será efectuada uma pesquisa se aceder ao iSeries através de um servidor de ficheiros. Se esta opção não estiver seleccionada, serão pesquisados todos os acessos.

- Falha no pedido se falhar o programa de saída - (\*ERRFAIL especificado)

Se existirem erros quando for chamado o programa de saída, o pedido ou operação que acionou a chamada do programa de saída falhará. Se esta opção não estiver seleccionada, o sistema ignorará a falha do programa de saída e o objecto será considerado como se não tivesse sido pesquisado.

- Executar actualizações do acesso para escrita - (\*NOWRTUPG não especificado)

A actualização do acesso ocorrerá para o descritor da pesquisa transferido para o programa de saída, de forma a incluir o acesso para escrita. Se a opção \*NOWRTUPG não for seleccionada, o sistema não efectuará tentativas de actualização do acesso para escrita.

Se a opção \*NOWRTUPG for especificada, o sistema **não** tentará actualizar o acesso para o descritor da pesquisa transferido para o programa de saída, de forma a incluir o acesso para escrita. Se a opção \*NOWRTUPG não for especificada, o sistema tentará efectuar a actualização do acesso para escrita.

- Utilize o atributo 'só quando os objectos foram alterados' para controlar a pesquisa - (\*USEOCOATR especificado)  
Será utilizado o atributo 'só alteração de objecto' (só pesquisará o objecto se este tiver sido modificado). Se esta opção não for seleccionada, este atributo não será utilizado e o objecto será pesquisado depois de modificado e assim que o software de pesquisa indicar uma actualização.
- Falha no pedido de encerramento se a pesquisa falhar durante o encerramento - (\*NOFAILCLO não especificado)  
Se ocorreu uma falha na pesquisa de um objecto durante o processamento de encerramento, o pedido de encerramento falhará. Se esta opção não for seleccionada, o pedido de encerramento não falhará. Quando não seleccionada, este valor substitui a especificação do valor 'falha no pedido se falhar o programa de saída'.  
Se a opção \*NOFAILCLO for especificada, **não** ocorrerá um falha no pedido de encerramento com indicação de falha na pesquisa, mesmo que tenha ocorrido uma falha na pesquisa do objecto efectuada como parte do processamento de encerramento.
- Pesquisar no próximo acesso após restauro do objecto - (\*NOPOSTRST não especificado)  
Os objectos serão pesquisados após o restauro. Se o atributo 'o objecto não será pesquisado' estiver especificado, o objecto será pesquisado uma vez após o restauro. Se o atributo 'só alteração de objecto' estiver especificado, o objecto será pesquisado após o restauro.  
Se a opção \*NOPOSTRST for especificada durante o restauro dos objectos, estes não serão pesquisados apenas porque foram restaurados. Se o atributo de objecto for 'o objecto não será pesquisado', o objecto nunca será pesquisado. Se o atributo de objecto for 'só alteração de objecto', o objecto só será pesquisado se for modificado após o restauro.

Para obter mais informações sobre estes dois valores do sistema, consulte QSCANFS ou QSCANFSCTL.

## Ocorrências de pesquisa

A pesquisa pode ocorrer por várias razões. A lista apresentada abaixo mostra quando e por que motivo poderá ocorrer uma pesquisa.

- "Alteração de objecto"
- "Alteração de assinatura"
- "CCSID diferente" na página 23
- "Durante uma operação de salvaguarda" na página 23
- "Verificar a integridade do objecto" na página 23

Para ver o estado da pesquisa actual e os atributos de um objecto, pode utilizar o comando Trabalhar com Ligações de Objecto (WRKLNK), a API Obter atributos (Qp0lGetAttr()) ou a página Properties do iSeries Navigator.

**Alteração de objecto:** Poderá ser efectuada uma pesquisa se o objecto for acedido após ter sido alterado ou modificado. Normalmente, a modificação ocorre nos dados do objecto. Modificações efectuadas a um objecto são, por exemplo, escrever no objecto directamente ou através da definição de memória, truncar ou eliminar o objecto. Se o atributo CCSID do objecto for alterado, será efectuada uma pesquisa no próximo acesso.

**Alteração de assinatura:** Será efectuada uma pesquisa quando o objecto for acedido se a assinatura global for diferente da assinatura do objecto. As assinaturas globais ou de grupo ASP independente representam o nível de software associado aos programas de saída relacionados com a pesquisa. Consulte QIBM\_QPOL\_SCAN\_OPEN para obter mais detalhes. A assinatura do objecto reflecte a assinatura global ou de ASP independente da última pesquisa efectuada ao objecto. Quando um objecto não se encontra num grupo de ASPs independentes, a assinatura do objecto é comparada com a assinatura de pesquisa global. Se o objecto se encontrar num ASP independente, a assinatura do objecto é comparada com a assinatura de pesquisa de grupo de ASPs independentes associado.

**Nota:** No exemplo que se segue, são utilizadas chaves de pesquisa de expressões e assinaturas chave de pesquisa. A chave de pesquisa é um método de identificação de um conjunto de software de pesquisa. Um exemplo disto é o de uma empresa específica. A assinatura chave de pesquisa permite que o conjunto de software de pesquisa indique o nível de suporte fornecido. Um exemplo disto é um conjunto de definições de vírus. Consulte QIBM\_QP0L\_SCAN\_OPEN para obter mais informações.

Segue-se um exemplo de uma situação em que um objecto não se encontra num grupo de ASPs independentes e é efectuada uma pesquisa:

1. Um programa de saída é registado no ponto de saída QIBM\_QP0L\_SCAN\_OPEN. Foram especificadas uma chave e uma assinatura chave de pesquisa da seguinte forma:

Chave de pesquisa: XXXXXX  
Assinatura chave de pesquisa: 0000000000

A assinatura de pesquisa global é 0000 e não está actualizada.

2. Um programa de saída é então registado no ponto de saída QIBM\_QP0L\_SCAN\_CLOSE. Foram especificadas uma chave e uma assinatura chave de pesquisa da seguinte forma:

Chave de pesquisa: XXXXXX  
Assinatura chave de pesquisa: 1111111111

A assinatura de pesquisa global é, em seguida, actualizada para 0001.

3. A seguir, é aberto um ficheiro que tem actualmente uma assinatura de objecto de 0000. A existência dos programas de saída, juntamente com a diferença de assinaturas de pesquisa globais (0000 a 0001), inicia uma pesquisa. Depois de concluída a pesquisa com êxito, a assinatura do ficheiro é actualizada para 0001.
4. Se o ficheiro for aberto por outro utilizador, não será efectuada nova pesquisa, uma vez que as assinaturas do objecto e as assinaturas globais correspondem.

O exemplo abaixo mostra que o programa de saída pretende executar uma nova pesquisa:

1. Foi adicionado suporte ao sistema para verificar a existência de novos tipos de vírus. A API Assinatura de Pesquisa Alterada, QP0LCHSG, é chamada para actualizar a assinatura chave de pesquisa das chaves de pesquisa. Uma chave e uma assinatura chave de pesquisa são especificadas da seguinte forma:

Chave de pesquisa: XXXXXX  
Assinatura chave de pesquisa: 2222222222

A assinatura chave de pesquisa global é, em seguida, actualizada para 0002.

2. Se o ficheiro pesquisado anteriormente for agora aberto, a diferença entre as assinaturas provocará uma nova pesquisa.

A seguir, podemos ver o exemplo de quando um objecto se encontra num grupo de ASPs independentes:

1. Um ASP independente é activado pela primeira vez e um ficheiro existente no ASP independente é aberto. Quando é aberto o primeiro ficheiro, a lista de chaves de pesquisa de ASP independente é comparada com a lista de chaves de pesquisa do sistema. As duas são diferentes devido ao facto de não existir uma lista de chaves de pesquisa do ASP independente. Neste caso, a lista de chaves de pesquisa do ASP independente obtém a lista de chaves de pesquisa globais. A lista de chaves de pesquisa do ASP independente terá então uma chave de pesquisa XXXXXX e uma assinatura chave de pesquisa 2222222222. Assim, a assinatura de pesquisa de ASP independente é alterada para 0001. Quando o ficheiro do ASP independente for aberto e tiver uma assinatura de objecto 0000, esta é

comparada com a assinatura de pesquisa de ASP independente 0001 e, devido à diferença entre elas, é efectuada uma pesquisa ao ficheiro. Depois de efectuada a pesquisa com êxito, a assinatura do ficheiro é actualizada para 0001.

**Nota:** Uma alteração na assinatura accionará uma pesquisa, a menos que o objecto tenha o atributo 'só alteração de objecto' e o valor de sistema \*USEOCOATR especificados.

**CCSID diferente:** Se um objecto for acedido com um CCSID diferente do pesquisado anteriormente, é accionada uma pesquisa.

Um exemplo disto é quando um ficheiro com dados armazenados no CCSID 819 é aberto no CCSID 1200 e pesquisado com êxito. Desde que os dados do ficheiro não sejam alterados, sempre que abrir o ficheiro no CCSID 1200, não é efectuada uma pesquisa. No entanto, se esse ficheiro for aberto num CCSID diferente, por exemplo, 37, é accionada uma pesquisa para o CCSID 37. Se também essa pesquisa for efectuada com êxito, qualquer acesso subsequente com o CCSID 1200 e 37 não accionará uma pesquisa adicional.

São apenas mantidos dois CCSIDs e uma indicação binária numa tentativa de reduzir os dados armazenados no sistema. Se normalmente aceder ao mesmo objecto com CCSIDs diferentes, poderão ser efectuadas várias pesquisas adicionais.

**Durante uma operação de salvaguarda:** Esta situação é ainda um outro exemplo de quando pode ocorrer uma pesquisa. Uma pesquisa pode ser pedida quando um objecto é guardado. O comando SAV agora inclui um parâmetro SCAN que permite especificar se os ficheiros serão pesquisados ao serem guardados. Também pode solicitar que o objecto não seja guardado se anteriormente falhou numa pesquisa ou se falhar durante uma pesquisa ao guardar. Deste modo, evita que os ficheiros que falharam na pesquisa sejam colocados nos suportes de dados e possivelmente movidos para outros sistemas.

**Nota:** Isto não significa que, quando restaurado o objecto, seja assinalado como pesquisado. Sempre que os objectos são restaurados, é eliminado o histórico do estado de pesquisa.

**Verificar a integridade do objecto:** Por último, pode ser solicitada uma pesquisa se o parâmetro SCANFS do comando Verificar Integridade do Objecto (CHKOBJITG) for especificado juntamente com um valor \*YES. Esta opção será a ideal se pretende determinar se um ficheiro está em bom estado sem ter de o abrir. Se SCANFS (\*Status) for especificado, todos os objectos que falharam nas pesquisas anteriores terão um registo de violação de falha de pesquisa.

---

## Trabalhar com sistemas de ficheiros

Um **sistema de ficheiros** fornece-lhe o suporte para aceder a segmentos específicos de armazenamento que estão organizados como unidades lógicas. Estas unidades lógicas no servidor são ficheiros, directórios, bibliotecas e objectos.

Cada sistema de ficheiros tem um conjunto de estruturas e regras lógicas para interagir com as informações em memória. Estas estruturas e regras podem ser diferentes de um sistema de ficheiros para outro. De facto, na perspectiva das estruturas e regras, o suporte do OS/400 para aceder a ficheiros de base de dados e vários outros tipos de objectos através de bibliotecas pode ser considerado como um sistema de ficheiros. Do mesmo modo, o suporte do OS/400 para aceder a documentos (que são, na realidade, ficheiros de dados contínuos) através da estrutura de pastas pode ser considerado como um sistema de ficheiros separado.

O sistema de ficheiros integrado considera o suporte de bibliotecas e o suporte de pastas como sistemas de ficheiros separados. Outros tipos de suporte de gestão de ficheiros que têm capacidades diferentes são também considerados como sistemas de ficheiros separados.

Para ver uma comparação das funções e limitações de cada sistema de ficheiros, consulte "Comparação entre sistemas de ficheiros" na página 25.

Os sistemas de ficheiros no sistema de ficheiros integrado são:

- "raiz" (/)
- Sistema de ficheiros de sistemas abertos (QOpenSys)
- Sistema de ficheiros definido pelo utilizador (UDFS)
- Sistema de ficheiros de biblioteca (QSYS.LIB)
- QSYS.LIB de ASP independente
- Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)
- Sistema de Ficheiros Óptico (QOPT)
- Sistema de ficheiros NetWare (QNetWare)
- Sistema de ficheiros do iSeries NetClient (QNTC)
- Sistema de ficheiros do Servidor de Ficheiros do OS/400 (QFileSvr.400)
- Sistema de ficheiros da rede (NFS)

Podem interagir com qualquer um dos sistemas de ficheiros através de uma interface comum. Esta interface é otimizada para o input e output de dados de sequência, ao contrário do input e output de registos, que é fornecido através das interfaces de gestão de dados. Os comandos, menus, ecrãs e as interfaces de programas de aplicações (APIs) fornecidos permitem a interacção com os sistemas de ficheiros através desta interface comum.

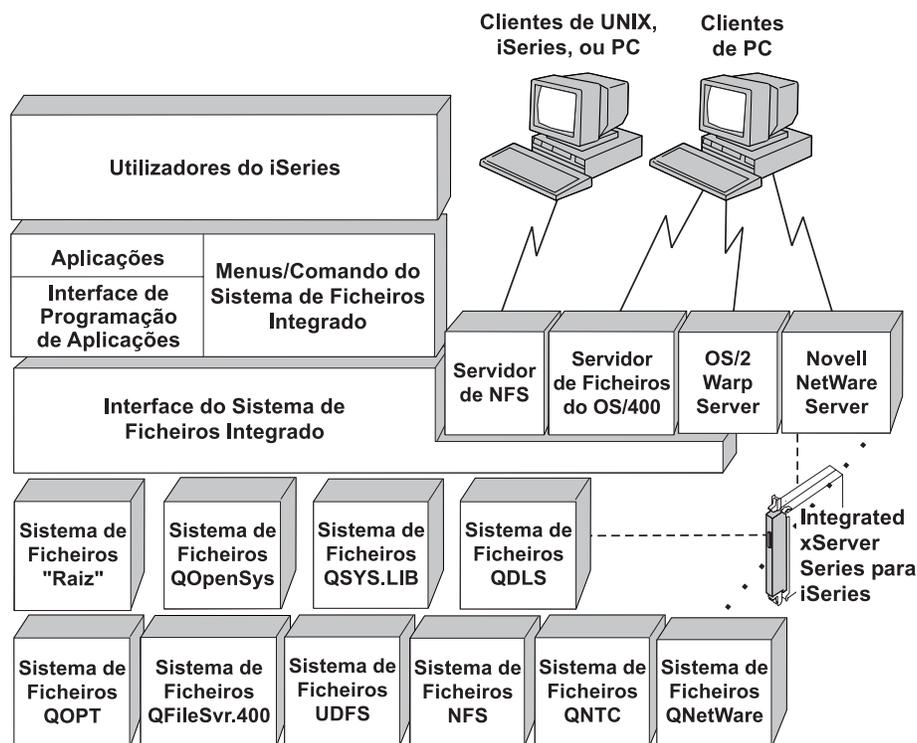


Figura 9. Sistemas de ficheiros, servidores de ficheiros e a interface do sistema de ficheiros integrado

Para obter mais informações, consulte os tópicos e as publicações seguintes:

- Suporte Óptico 
- Network File System Support OS/400 

## Comparação entre sistemas de ficheiros

A Tabela 2 e a Tabela 3 na página 26 resumem as funções e limitações de cada sistema de ficheiros.

Tabela 2. Resumo dos Sistemas de Ficheiros (Parte 1 de 2)

Capacidade	"raiz" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
Parte standard do OS/400	Sim	Sim	Sim	Sim	Sim
Tipo de ficheiro	Dados Contínuos	Dados Contínuos	Registo <sup>12</sup>	Dados Contínuos	Dados Contínuos
Limite do tamanho do ficheiro	T2=1 TB; T1=256 GB	T2=1 TB; T1=256 GB	Tamanhos dos ficheiros de base de dados	4 GB	Varia <sup>17</sup>
Integrado com o OfficeVision® (por exemplo, o ficheiro pode ser enviado por correio)	Não	Não	Não	Sim	Não
Acesso através do servidor de ficheiros do OS/400	Sim	Sim	Sim	Sim	Sim
Acesso directo através do file server I/O processor <sup>1</sup>	Não	Não	Não	Não	Sim
Velocidade comparativa para abrir/fechar	Média <sup>2</sup>	Média <sup>2</sup>	Baixa <sup>2</sup>	Baixa <sup>2</sup>	Média <sup>2</sup>
Procura de nomes sensível a maiúsculas e minúsculas	Não	Sim	Não <sup>4</sup>	Não <sup>5</sup>	Não
Comprimento máximo de cada componente no nome de caminho	255 car	255 car	10,6 car <sup>6</sup>	8,3 car <sup>7</sup>	255 car
Comprimento máximo do nome de caminho <sup>8</sup>	16MB	16MB	55 – 66 car <sup>4</sup>	82 car	255 car
Comprimento máximo dos atributos expandidos para um objecto	2GB	2GB	Varia <sup>9</sup>	32KB	0 <sup>18</sup>
Níveis máximos da hierarquia de directórios dentro do sistema de ficheiros	Sem limite <sup>10</sup>	Sem limite <sup>10</sup>	3	32	127
Máximo de ligações por objecto <sup>11</sup>	Varia <sup>15</sup>	Varia <sup>15</sup>	1	1	1
Suporta ligações simbólicas	Sim	Sim	Não	Não	Não
Objecto/ficheiro pode ter proprietário	Sim	Sim	Sim	Sim	Não
Suporta comandos do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta APIs do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta APIs do sistema de ficheiros hierárquico (HFS)	Não	Não	Não	Sim	Não
Protecção por módulo <sup>13</sup>	Sim	Sim	Sim	Não	Sim
Suporta registo em diário de objectos	Sim	Sim	Sim <sup>14</sup>	Não	Não

Tabela 2. Resumo dos Sistemas de Ficheiros (Parte 1 de 2) (continuação)

Capacidade	"raiz" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
<p><b>Notas:</b></p> <ol style="list-style-type: none"> <li>O file server I/O processor é hardware utilizado pelo LAN Server.</li> <li>Quando acedido através do servidor de ficheiros do OS/400.</li> <li>Quando acedido através de um PC cliente do LAN Server. O acesso utilizando as APIs do iSeries é comparativamente lento.</li> <li>O sistema de ficheiros QSYS.LIB tem um comprimento de nome de caminho máximo de 55 caracteres. Consulte "Sistema de ficheiros de biblioteca (QSYS.LIB)" na página 37 para obter detalhes. O sistema de ficheiros QSYS.LIB de ASP independente tem um comprimento de caminho máximo de 66 caracteres. Consulte "QSYS.LIB de ASP independente" na página 40 para obter detalhes.</li> <li>Consulte "Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)" na página 43 para obter detalhes.</li> <li>Até 10 caracteres para o nome do objecto e até 6 caracteres para o tipo do objecto. Consulte "Sistema de ficheiros de biblioteca (QSYS.LIB)" na página 37 para obter mais detalhes.</li> <li>Até 8 caracteres para o nome e de 1 a 3 caracteres para a extensão do tipo de ficheiro (se existir alguma). Consulte "Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)" na página 43 para obter detalhes.</li> <li>Assumindo um nome de caminho absoluto que começa por / seguido do nome do sistema de ficheiros (tal como /QDLS...).</li> <li>Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente suportam três atributos expandidos predefinidos: .SUBJECT, .CODEPAGE e .TYPE. O comprimento máximo é determinado pelo comprimento combinado destes três atributos expandidos.</li> <li>Na prática, os níveis de directórios são limitados pelos limites de espaço do programa e do sistema.</li> <li>Excepto um directório, que pode ter apenas uma ligação a outro directório.</li> <li>Os espaços de utilizador no sistema de ficheiros QSYS.LIB e QSYS.LIB de ASP independente suportam input e output de ficheiros de dados contínuos.</li> <li>As APIs do sistema de ficheiros integrado têm protecção para módulos quando a operação é direccionada para um objecto que reside num sistema de ficheiros com protecção para módulos. Quando estas APIs são executadas em objectos de sistemas de ficheiros sem protecção para módulos quando estiverem a ser executados múltiplos módulos no trabalho, a API falhará.</li> <li>Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente suportam o registo em diário de tipos de objectos diferentes dos sistemas de ficheiros "raiz" (/), UDFS e QOpenSys. Consulte o tópico Journal management do iSeries Information Center para obter mais informações sobre o registo em diário de objectos residentes nos sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente.</li> <li>Os directórios *TYPE2 têm um limite de um milhão de ligações por objecto e um limite de 999.998 subdirectórios. Os directórios *TYPE1 têm um limite de 32.767 ligações por objecto. Para obter mais informações, consulte Directórios *TYPE2.</li> <li>Os dados desta coluna referem-se a ambos os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente.</li> <li>Depende do sistema a ser acedido</li> <li>O QNTC não suporta atributos expandidos</li> </ol> <p><b>Abreviaturas</b></p> <p>car = caracteres  T1 = *TYPE1 *STMF  T2 = *TYPE2 *STMF  B = bytes    KB = quilobytes    MB = megabytes    GB = gigabytes    TB = terabytes</p>					

Tabela 3. Resumo do Sistema de Ficheiros (Parte 2 de 2)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Parte standard do OS/400	Sim	Sim	Sim	Sim	Não

Tabela 3. Resumo do Sistema de Ficheiros (Parte 2 de 2) (continuação)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Tipo de ficheiro	Dados Contínuos	Dados Contínuos	Dados Contínuos	Dados Contínuos	Dados Contínuos
Limite do tamanho do ficheiro	4 GB	4 GB	T2 = 1 TB; T1=256 GB	Varia <sup>16</sup>	
Integrado com o OfficeVision (por exemplo, o ficheiro pode ser enviado por correio)	Não	Não	Não	Não	Não
Acesso através do servidor de ficheiros do OS/400	Sim	Sim	Sim	Sim	Sim
Acesso directo através do Integrated PC Server <sup>1</sup>	Não	Não	Não	Não	Sim
Velocidade comparativa para abrir/fechar	Baixa	Baixa <sup>2</sup>	Média <sup>2</sup>	Média <sup>2</sup>	Alta <sup>11</sup>
Procura de nomes sensível a maiúsculas e minúsculas	Não	Não <sup>2</sup>	Sim <sup>12</sup>	Varia <sup>2</sup>	Não
Comprimento máximo de cada componente no nome de caminho	Varia <sup>4</sup>	Varia <sup>2</sup>	255 car	Varia <sup>2</sup>	255 car <sup>13</sup>
Comprimento máximo do nome de caminho	294 car	Sem limite <sup>2</sup>	16MB	Sem limite <sup>2</sup>	255 car
Comprimento máximo dos atributos expandidos para um objecto	8MB	0 <sup>6</sup>	2GB <sup>10</sup>	0 <sup>6</sup>	64KB
Níveis máximos da hierarquia de directórios dentro do sistema de ficheiros	Sem limite <sup>7</sup>	Sem limite <sup>2</sup>	Sem limite <sup>7</sup>	Sem limite <sup>2</sup>	100
Máximo de ligações por objecto <sup>7</sup>	1	1	Varia <sup>15</sup>	Varia <sup>2</sup>	1
Suporta ligações simbólicas	Não	Não	Sim	Sim <sup>2</sup>	Não
Objecto/ficheiro pode ter proprietário	Não	Não <sup>9</sup>	Sim	Sim <sup>2</sup>	Sim
Suporta comandos do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta APIs do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta APIs do sistema de ficheiros hierárquico (HFS)	Sim	Não	Não	Não <sup>2</sup>	Não
Protecção por módulo <sup>14</sup>	Sim	Sim	Sim	Sim	Não
Suporta registo em diário de objectos	Não	Não	Sim	Não	Não

Tabela 3. Resumo do Sistema de Ficheiros (Parte 2 de 2) (continuação)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
<b>Notas:</b>					
<ol style="list-style-type: none"> <li>1. O file server I/O processor é hardware utilizado pelo LAN Server.</li> <li>2. Depende em que sistema de ficheiros está a ser acedido.</li> <li>3. Quando acedido através do servidor de ficheiros do OS/400</li> <li>4. Consulte "Sistema de Ficheiros Óptico (QOPT)" na página 45 para obter detalhes.</li> <li>5. Assumindo um nome de caminho absoluto que começa por uma / seguido do nome do sistema de ficheiros.</li> <li>6. O sistema de ficheiros QFileSvr.400 não devolve atributos expandidos mesmo que o sistema de ficheiros a ser acedido suporte atributos expandidos.</li> <li>7. Na prática, os níveis de directórios são limitados pelos limites de espaço do programa e do sistema.</li> <li>8. Excepto um directório, que pode ter apenas uma ligação a outro directório.</li> <li>9. O sistema de ficheiros a ser acedido poderá suportar proprietários de objecto.</li> <li>10. O comprimento máximo dos atributos expandidos para o próprio UDFS não pode exceder 40 bytes.</li> <li>11. Quando acedido através de um PC cliente da Novell NetWare. O acesso utilizando as APIs do iSeries é comparativamente lento.</li> <li>12. A opção sensível a maiúsculas e minúsculas pode ser especificada quando for criado um UDFS. Se o parâmetro *MIXED for utilizado quando criar um UDFS, permitirá uma procura sensível a maiúsculas e minúsculas.</li> <li>13. Os objectos de Serviços de Directório da NetWare têm um máximo de 255 caracteres. Os ficheiros e os directórios estão limitados ao formato 8.3 do DOS.</li> <li>14. As APIs do sistema de ficheiros integrado estão protegidas por módulo quando são acedidas num processo suportando múltiplos módulos. O sistema de ficheiros não permite o acesso aos sistemas de ficheiros que não estejam protegidos por módulos.</li> <li>15. Os directórios *TYPE2 têm um limite de um milhão de ligações por objecto. Os directórios *TYPE1 têm um limite de 32.767 ligações por objecto. Para obter mais informações, consulte Directórios *TYPE2.</li> <li>16. Depende do sistema a ser acedido</li> </ol>					
<b>Abreviaturas</b>					
car = caracteres					
T1 = *TYPE1 *STMF					
T2 = *TYPE2 *STMF					
B = bytes   KB = kilobytes   MB = megabytes   GB = gigabytes   TB = terabytes					

## Sistema de ficheiros "raiz" (/)

O sistema de ficheiros "raiz" (/) tira o máximo partido do suporte de ficheiros de dados contínuos e da estrutura de directórios hierárquica do sistema de ficheiros integrado. O sistema de ficheiros "raiz" (/) tem as características dos sistemas de ficheiros do DOS (Disk Operating System) e do OS/2.

Para além disso:

- Está optimizado para input e output de ficheiros de dados contínuos.
- Suporta múltiplas ligações permanentes e ligações simbólicas.
- Suporta sockets locais.
- Suporta interfaces de programas de aplicações (APIs) com protecção para módulos.
- Suporta objectos \*FIFO.
- Suporta os objectos /dev/null e /dev/zero \*CHRSE, bem como outros objectos \*CHRSE.
- Suporta o registo em diário de alterações a objectos.
- Suporta a pesquisa de objectos através da utilização dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado.

O sistema de ficheiros "raiz" (/) tem suporte para os ficheiros especiais de caracteres (\*CHRSF) chamados /dev/null e /dev/zero. Os ficheiros especiais de caracteres estão associados a um dispositivo ou um recurso de um sistema informático. Têm nomes de caminho que aparecem em directórios e têm a mesma protecção de acesso que os ficheiros normais. Os ficheiros especiais de caracteres /dev/null ou /dev/zero estão sempre vazios e quaisquer dados escritos em /dev/null ou /dev/zero serão eliminados. Os ficheiros /dev/null e /dev/zero têm um tipo de objecto \*CHRSF e podem ser utilizados como ficheiros normais, excepto quanto ao facto de não serem lidos dados do ficheiro /dev/null e de o ficheiro /dev/zero ser sempre devolvido com êxito, sendo os dados convertidos em zeros.

Para obter mais informações sobre o sistema de ficheiros "raiz" (/), consulte Utilizar o sistema de ficheiros "raiz" (/).

### **Utilizar o sistema de ficheiros "raiz" (/)**

O sistema de ficheiros "raiz" (/) pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs do utilizador e APIs do sistema de ficheiros integrado.

- "Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros "raiz" (/)"
- "Nomes de caminho no sistema de ficheiros "raiz" (/)"
- "Ligações no sistema de ficheiros "raiz" (/)"
- "Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros "raiz" (/)"
- "Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros "raiz" (/)" na página 30
- "Registar em diário alterações a objectos no sistema de ficheiros "raiz" (/)" na página 30
- "Dispositivos UDP e TCP no sistema de ficheiros "raiz" (/)" na página 30

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros "raiz" (/):** O sistema de ficheiros mantém o mesmo formato de maiúsculas e minúsculas no qual os nomes do objecto são introduzidos, mas não é feita qualquer distinção entre maiúsculas e minúsculas quando o servidor procura nomes.

### **Nomes de caminho no sistema de ficheiros "raiz" (/):**

- Os nomes de caminho têm o seguinte formato:  
Directório/Directório . . . /Objecto
- Cada componente do nome de caminho pode ter até 255 caracteres de comprimento, muito mais do que nos sistemas de ficheiros QSYS.LIB ou QDLS. O nome completo do caminho pode ser extremamente longo, até 16 megabytes.
- Não existe limite para a profundidade da hierarquia de directórios, para além dos limites de espaço do programa e servidor.
- Os caracteres dos nomes são convertidos no formato UCS2 Nível 1 (para directórios \*TYPE1) e UTF-16 (para directórios \*TYPE2) quando os nomes são armazenados (consulte "Continuidade de nomes" na página 17). Consulte Directórios \*TYPE2 para obter mais informações sobre os formatos de directório.

**Ligações no sistema de ficheiros "raiz" (/):** São permitidas múltiplas ligações permanentes para o mesmo objecto no sistema de ficheiros "raiz" (/). As ligações simbólicas são completamente suportadas. Uma ligação simbólica pode ser utilizada para estabelecer ligação entre o sistema de ficheiros "raiz" (/) e um objecto de outro sistema de ficheiros como, por exemplo, QSYS.LIB, QSYS.LIB de ASP independente ou QDLS.

Para obter uma descrição de ligações, consulte "Ligação" na página 11.

**Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros "raiz" (/):** Todos os comandos incluídos em "Aceder através de comandos de CL" na página 62 e os ecrãs descritos em "Aceder através de menus e ecrãs" na página 61 podem funcionar no sistema de ficheiros "raiz" (/). No entanto, a utilização destes comandos num processo que suporte múltiplos módulos pode não ser segura.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros "raiz" (/):** Todas as APIs incluídas em "Executar operações utilizando APIs" na página 98 podem ser utilizadas no sistema de ficheiros "raiz" (/). Consulte APIs para obter mais informações sobre APIs, como a protecção por módulos.

**Registar em diário alterações a objectos no sistema de ficheiros "raiz" (/):** Os objectos do sistema de ficheiros "raiz" (/) podem ser registados em diário. O principal objectivo da gestão de registos em diário consiste em permitir que o utilizador recupere as alterações ocorridas num objecto, desde a última vez que este foi guardado. Para obter mais informações sobre o registo em diário de alterações a objectos no sistema de ficheiros "raiz" (/), consulte "Registar objectos em diário" na página 88.

**Dispositivos UDP e TCP no sistema de ficheiros "raiz" (/):** O sistema de ficheiros "raiz" (/) no directório /dev/xti não suportará dois controladores de dispositivos com o nome udp e tcp. Estes dois controladores são ficheiros especiais de caracteres (\*CHRSEFs) e serão criados durante o primeiro carregamento do programa inicial (IPL). Os controladores de dispositivos udp e tcp serão utilizados para abrir uma ligação para os fornecedores de transporte UDP e TCP. Estes dois controladores serão dispositivos de utilizador e receberão um número principal de dispositivo novo. Eles também terão aberturas idênticas, o que significa que, cada abertura obterá uma instância exclusiva do dispositivo. A utilização destes dispositivos só será suportada em PASE (Portable Application Solutions Environment). A Tabela 4 anterior contém os objectos que serão criados e as respectivas propriedades.

Tabela 4. Objectos e propriedades do controlador de dispositivos

Nome de caminho	Tipo	Principal	Secundário	Proprietário	Autoridades sobre os dados do proprietário	Grupo	Autoridades sobre os dados do grupo	Autoridades sobre os dados públicos
/dev/xti	*DIR	N/A	N/A	QSYS	*RWX	Nenhum	*RX	*RX
/dev/xti/tcp	*CHRSEF	Conjunto de unidades	TCP	QSYS	*RW	Nenhum	*RW	*RW
/dev/xti/udp	*CHRSEF	Conjunto de unidades	UDP	QSYS	*RW	Nenhum	*RW	*RW

## Sistema de ficheiros de sistemas abertos (QOpenSys)

O sistema de ficheiros QOpenSys é compatível com as normas do sistema aberto baseado em UNIX, como POSIX e XPG. À semelhança do sistema de ficheiros "raiz" (/), este sistema de ficheiros tira partido do suporte de ficheiros de dados contínuos e de directórios, fornecido pelo sistema de ficheiros integrado.

Para além disso:

- É acedido através de uma estrutura de directórios hierárquica, semelhante à dos sistemas UNIX.
- Está optimizado para input e output de ficheiros de dados contínuos.
- Suporta múltiplas ligações permanentes e ligações simbólicas.
- Suporta nomes sensíveis a maiúsculas e minúsculas.
- Suporta sockets locais.
- Suporta APIs com protecção para módulos.
- Suporta objectos \*FIFO.
- Suporta o registo em diário de alterações a objectos.
- Suporta a pesquisa de objectos através da utilização dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado.

O sistema de ficheiros QOpenSys tem as mesmas características que o sistema de ficheiros "raiz" (/), excepto no que diz respeito à sensibilidade a maiúsculas e minúsculas em relação à activação do suporte das normas de sistemas abertos baseados em UNIX.

Para obter mais informações sobre o QOpenSys, consulte Utilizar o QOpenSys.

## Utilizar o QOpenSys

O sistema de ficheiros QOpenSys pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs do utilizador e APIs do sistema de ficheiros integrado.

- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QOpenSys”
- “Nomes de caminho no sistema de ficheiros QOpenSys”
- “Ligações no sistema de ficheiros QOpenSys”
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys”
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys”
- “Registar em diário alterações a objectos no sistema de ficheiros QOpenSys”

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QOpenSys:** Ao contrário do sistema de ficheiros “raiz” (/), o sistema de ficheiros QOpenSys faz a distinção entre maiúsculas ou minúsculas ao procurar nomes de objectos. Por exemplo, uma cadeia de caracteres em maiúsculas não corresponderá à mesma cadeia de caracteres em que qualquer carácter seja minúsculo.

Esta sensibilidade a maiúsculas e minúsculas permite-lhe utilizar nomes duplicados, se houver qualquer diferença nas maiúsculas e minúsculas dos caracteres que constituem o nome. Por exemplo, pode ter acesso a um objecto denominado Salários, a um objecto denominado Salarios e a um objecto denominado SALÁRIOS no mesmo directório do QOpenSys.

### Nomes de caminho no sistema de ficheiros QOpenSys:

- Os nomes de caminho têm o seguinte formato:  
Directório/Directório/ . . . /Objecto
- Cada componente do nome de caminho pode ter até 255 caracteres de comprimento. O nome completo do caminho pode ter até 16 megabytes de comprimento.
- Não existe limite para a profundidade da hierarquia de directórios, para além dos limites de espaço do programa e servidor.
- Os caracteres dos nomes são convertidos no formato UCS2 Nível 1 (para directórios \*TYPE1) e UTF-16 (para directórios \*TYPE2) quando os nomes são armazenados (consulte “Continuidade de nomes” na página 17). Consulte Directórios \*TYPE2 para obter mais informações sobre os formatos de directório.

**Ligações no sistema de ficheiros QOpenSys:** São permitidas múltiplas ligações permanentes para o mesmo objecto no sistema de ficheiros QOpenSys. As ligações simbólicas são completamente suportadas. Uma ligação simbólica pode ser utilizada para ligar um sistema de ficheiros QOpenSys a um objecto de outro sistema de ficheiros.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys:** Todos os comandos incluídos em “Aceder através de comandos de CL” na página 62 e os ecrãs descritos em “Aceder através de menus e ecrãs” na página 61 podem funcionar no sistema de ficheiros QOpenSys. No entanto, a utilização destes comandos num processo que suporte múltiplos módulos pode não ser segura.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys:** Todas as APIs incluídas em “Executar operações utilizando APIs” na página 98 podem ser utilizadas no sistema de ficheiros QOpenSys. Consulte APIs para obter mais informações sobre APIs, como a protecção por módulos.

**Registar em diário alterações a objectos no sistema de ficheiros QOpenSys:** Os objectos no sistema de ficheiros QOpenSys podem ser registados em diário. O principal objectivo da gestão de registos em diário consiste em permitir que o utilizador recupere as alterações ocorridas num objecto, desde a última vez

que este foi guardado. Para obter mais informações sobre o registo em diário de alterações a objectos no sistema de ficheiros QOpenSys, consulte “Registar objectos em diário” na página 88.

## Sistema de ficheiros definido pelo utilizador (UDFS)

Os sistemas de ficheiros UDFS residem no conjunto de memória auxiliar (ASP) ou num conjunto de memória auxiliar (ASP) independente à sua escolha. Pode criar e gerir estes sistemas de ficheiros.

Além disso, estes sistemas:

- Fornecem uma estrutura de directórios hierárquica, semelhante à dos sistemas operativos de PC, tal como DOS e OS/2.
- Estão optimizados para input e output de ficheiros de dados contínuos.
- Suportam múltiplas ligações permanentes e ligações simbólicas.
- Suportam sockets locais.
- Suportam APIs com protecção contra módulos.
- Suportam objectos \*FIFO.
- Suportam o registo em diário de alterações a objectos.
- Suportam a pesquisa de objectos através dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado.

Para obter mais informações sobre a criação de UDFSs, consulte CRTUDFS. Pode criar múltiplos UDFSs atribuindo a cada um nome exclusivo. Pode especificar outros atributos para um UDFS durante a criação, incluindo:

- Um número de ASP ou nome de ASP independente onde são armazenados os objectos que estão localizados no UDFS.
- As características de sensibilidade a maiúsculas e minúsculas dos nomes de objecto localizados num UDFS.

A sensibilidade a maiúsculas e minúsculas de um UDFS determina se os caracteres em maiúsculas e minúsculas terão correspondência quando procurar nomes de objecto no UDFS.

- O atributo de pesquisa de objectos criados que define qual o atributo de pesquisa para objectos criados num UDFS. Para obter informações sobre a definição deste atributo, consulte “Definir se os objectos devem ou não ser pesquisados” na página 119.
- O atributo de restrição de mudança de nome e de remoção.
- O valor de auditoria para um UDFS.
- Os diferentes formatos de ficheiros de dados contínuos, \*TYPE1 e \*TYPE2. Para obter informações adicionais sobre estes dois formatos de ficheiros de dados contínuos, consulte “Ficheiro de dados contínuos” na página 15.

Consulte os tópicos que se seguem para obter mais informações sobre sistemas de ficheiros definidos pelo utilizador:

- Conceitos de UDFS
- Utilizar o UDFS através da interface do sistema de ficheiros integrado

## Conceitos de UDFS

Num UDFS, tal como nos sistemas de ficheiros “raiz” (/) e QOpenSys, pode criar directórios, ficheiros de dados contínuos, ligações simbólicas, sockets locais e objectos \*FIFO.

Um objecto ficheiro especial de um só bloco (\*BLKSF) representa um UDFS. À medida que cria UDFSs, também cria automaticamente ficheiros especiais de bloco. O utilizador só pode ter acesso ao ficheiro especial de bloco utilizando os comandos genéricos do sistema de ficheiros integrado, das APIs e da interface do QFileSvr.400.

Um UDFS existe apenas em dois estados: **instalado** e **desinstalado**. Quando instala um UDFS, os objectos nele contidos ficam acessíveis. Quando desinstala um UDFS, os objectos nele contidos ficam inacessíveis.

Para aceder a objectos de um UDFS, terá de instalar o UDFS num directório (por exemplo, /home/JOSE). Quando instala um UDFS num directório, o conteúdo original desse directório, incluindo objectos e subdirectórios, fica inacessível. Quando instala um UDFS, o seu conteúdo fica acessível através do caminho do directório sobre o qual instalou o UDFS. Por exemplo, o directório /home/JOSE contém um ficheiro /home/JOSE/salarios. Um UDFS contém três directórios correio, accao e enviar. Depois de instalar o UDFS em /home/JOSE, o ficheiro /home/JOSE/salarios fica inacessível e os três directórios do UDFS ficam acessíveis como /home/JOSE/correio, /home/JOSE/accao e /home/JOSE/enviar. Depois de desinstalar o UDFS, o ficheiro /home/JOSE/salarios fica de novo acessível e os três directórios do UDFS ficam inacessíveis.

**Nota:** Não pode instalar um UDFS por cima de outro num ASP independente.

Para obter mais informações sobre a instalação de sistemas de ficheiros, consulte OS/400 Network File System Support .

### **Utilizar o UDFS através da interface do sistema de ficheiros integrado**

Um UDFS pode ser acedido através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros OS/400 ou os comandos do sistema de ficheiros integrado, ecrãs de utilizador e APIs. Ao utilizar a interface do sistema de ficheiros integrado, deverá ter em conta as considerações e limitações que se seguem.

- “Sensibilidade a maiúsculas e minúsculas num UDFS do sistema de ficheiros integrado”
- “Nomes de caminho num UDFS do sistema de ficheiros integrado” na página 34
- “Ligações num UDFS do sistema de ficheiros integrado” na página 34
- “Utilizar comandos do sistema de ficheiros integrado num UDFS” na página 34
- “Utilizar APIs do sistema de ficheiros integrado num UDFS” na página 35
- “Interface gráfica do utilizador para um UDFS” na página 35
- “Criar um UDFS do sistema de ficheiros integrado” na página 35
- “Eliminar um UDFS do sistema de ficheiros integrado” na página 35
- “Ver um UDFS do sistema de ficheiros integrado” na página 36
- “Instalar um UDFS do sistema de ficheiros integrado” na página 36
- “Desinstalar um UDFS do sistema de ficheiros integrado” na página 36
- “Guardar e restaurar um UDFS do sistema de ficheiros integrado” na página 36
- “Registar em diário alterações a objectos num sistema de ficheiros UDFS” na página 36
- “Conjuntos de Memória Auxiliar (ASPs) Independentes e UDFS” na página 36

**Sensibilidade a maiúsculas e minúsculas num UDFS do sistema de ficheiros integrado:** Pode especificar se os nomes de objecto contidos no UDFS serão ou não sensíveis a maiúsculas e minúsculas quando este é criado.

Quando activa (a opção) sensibilidade a maiúsculas e minúsculas, os caracteres maiúsculos e minúsculos serão distinguidos quando procurar nomes de objecto. Por exemplo, um nome que esteja totalmente em maiúsculas não corresponderá ao mesmo nome, se este estiver todo em minúsculas. Deste modo, /home/MARIA/ e /home/maria/ são reconhecidos como directórios diferentes. Para criar um UDFS sensível a maiúsculas e minúsculas, pode especificar \*MIXED para o parâmetro CASE, quando utilizar o comando CRTUDFS.

Quando activa (a opção) insensibilidade a maiúsculas e minúsculas, o servidor não distinguirá entre maiúsculas e minúsculas durante a procura de nomes. Deste modo, o servidor reconhecerá /home/CARLA

e /HOME/carla como o mesmo directório e não como dois directórios diferentes. Para criar um UDFS insensível a maiúsculas e minúsculas, pode especificar \*MONO para o parâmetro CASE quando utilizar o comando CRTUDFS.

Em qualquer dos casos, o sistema de ficheiros guarda os mesmos formatos de maiúsculas e minúsculas nos quais o utilizador introduziu os nomes de objecto. A opção de sensibilidade a maiúsculas e minúsculas só se aplica ao modo como o utilizador procura nomes no servidor.

**Nomes de caminho num UDFS do sistema de ficheiros integrado:** Um ficheiro especial de bloco (\*BLKSF) representa um UDFS quando é necessário manipular todo o UDFS e todos os objectos nele contidos. Se o seu UDFS residir no ASP do sistema ou num ASP base do utilizador, os nomes de ficheiros especiais de blocos têm de estar no formato

```
/disp/QASPXX/nome_udfs.udfs
```

onde XX é o número do ASP onde armazenou o UDFS e nome\_udfs é o nome exclusivo do UDFS nesse ASP. Tenha em atenção que o nome do UDFS tem de terminar com a extensão .udfs.

Se o seu UDFS residir num ASP independente, os nomes de ficheiros especiais de blocos têm de estar no formato

```
/dev/nome_asp/nome_udfs.udfs
```

em que nome\_asp é o nome do ASP independente onde vai armazenar o UDFS e nome\_udfs é o nome exclusivo do UDFS nesse ASP independente. Tenha em atenção que o nome do UDFS tem de terminar com a extensão .udfs.

Os nomes de caminho para objectos contidos num UDFS são relativos ao directório no qual um UDFS é instalado. Por exemplo, se instalar o UDFS /disp/qasp01/wysocki.udfs em /home/dinis, os nomes de caminho de todos os objectos do UDFS começarão por /home/dinis.

Regras de nomes de caminhos adicionais:

- Cada componente do nome de caminho pode ter até 255 caracteres de comprimento. O nome completo do caminho pode ter até 16 megabytes de comprimento.
- Não existe limite para a profundidade da hierarquia de directórios, para além dos limites de espaço do programa e servidor.
- Os caracteres dos nomes são convertidos no formato UCS2 Nível 1 (para directórios \*TYPE1) e UTF-16 (para directórios \*TYPE2) quando os nomes são armazenados (consulte “Continuidade de nomes” na página 17). Consulte Directórios \*TYPE2 para obter mais informações sobre os formatos de directório.

**Ligações num UDFS do sistema de ficheiros integrado:** Um UDFS permite múltiplas ligações permanentes ao mesmo objecto e suporta totalmente as ligações simbólicas. Uma ligação simbólica pode criar uma ligação de um UDFS a um objecto noutra sistema de ficheiros.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos do sistema de ficheiros integrado num UDFS:** Todos os comandos incluídos em “Aceder através de comandos de CL” na página 62 e os ecrãs descritos em “Aceder através de menus e ecrãs” na página 61 podem funcionar num sistema de ficheiros definido pelo utilizador. Existem alguns comandos de CL específicos para o sistema de ficheiros definido pelo utilizador e outros sistemas de ficheiros instalados em geral. A tabela seguinte descreve-os.

*Tabela 5. Comandos de CL do Sistema de Ficheiros Definido pelo Utilizador*

Comando	Descrição
ADDMFS	Adicionar Sistema de Ficheiros Instalado. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais.

Tabela 5. Comandos de CL do Sistema de Ficheiros Definido pelo Utilizador (continuação)

Comando	Descrição
CRTUDFS	Criar UDFS. Cria um sistema de ficheiros definido pelo utilizador.
DLTUDFS	Eliminar UDFS. Elimina um sistema de ficheiros definido pelo utilizador.
DSPMFSINF	Ver Informações sobre o Sistema de Ficheiros Instalado. Mostra informações sobre um sistema de ficheiros instalado.
DSPUDFS	Visualizar UDFS. Apresenta informações acerca de um sistema de ficheiros definido pelo utilizador.
MOUNT	Instalar um Sistema de Ficheiros. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais. Este comando é uma alternativa ao comando ADDMFS.
RMVMFS	Remover Sistema de Ficheiros Instalado. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local.
UNMOUNT	Desinstalar um Sistema de Ficheiros. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local. Este comando é alternativo ao comando RMVMFS.

**Nota:** Tem de instalar um UDFS para poder executar comandos do sistema de ficheiros integrado nos objectos armazenados nesse UDFS.

**Utilizar APIs do sistema de ficheiros integrado num UDFS:** Todas as APIs incluídas em “Executar operações utilizando APIs” na página 98 podem ser utilizadas num sistema de ficheiros definido pelo utilizador.

**Nota:** Tem de instalar um UDFS para as APIs do sistema de ficheiros integrado poderem executar operações nos objectos armazenados nesse UDFS.

**Interface gráfica do utilizador para um UDFS:** O iSeries Navigator, uma interface gráfica de utilizador no seu PC, fornece um acesso fácil e conveniente a UDFSs. Esta interface permite-lhe criar, eliminar, visualizar, instalar e desinstalar um UDFS de um cliente do Windows.

Pode executar operações num UDFS através do iSeries Navigator. Algumas tarefas básicas incluem:

- “Criar um UDFS do sistema de ficheiros integrado”.
- “Instalar um UDFS do sistema de ficheiros integrado” na página 36.
- “Desinstalar um UDFS do sistema de ficheiros integrado” na página 36.

**Criar um UDFS do sistema de ficheiros integrado:** O comando Criar Sistema de Ficheiros Definido Pelo Utilizador (CRTUDFS) cria um sistema de ficheiros que pode tornar-se visível através do espaço de nome, das APIs e dos comandos de CL do sistema de ficheiros integrado. Os comandos ADDMFS ou MOUNT colocam o UDFS “sobre” o directório local já existente. Pode criar um UDFS num ASP ou ASP independente que escolher.

Também pode especificar os seguintes itens para um UDFS:

- Sensibilidade a maiúsculas e minúsculas
- Definir se os objectos criados no UDFS devem ou não ser pesquisados
- O valor de auditoria para objectos criados no UDFS
- O valor para o atributo de mudança de nome e remoção restritos

**Eliminar um UDFS do sistema de ficheiros integrado:** O comando Eliminar Sistema de Ficheiros Definido pelo Utilizador (DLTUDFS) elimina um UDFS existente, por instalar, bem como todos os objectos nele contidos. O comando não será executado se o UDFS estiver instalado. A eliminação de um

UDFS causará a eliminação de todos os respectivos objectos. Se não tiver a autoridade adequada para eliminar todos os objectos de um UDFS, nenhum dos objectos será eliminado.

**Ver um UDFS do sistema de ficheiros integrado:** O comando Ver Sistema de Ficheiros Definido pelo Utilizador (DSPUDFS) apresenta os atributos de um UDFS existente, quer esteja instalado ou desinstalado. O comando Ver Informações sobre Sistemas de Ficheiros Instalados (DSPMFSINF) também apresentará informações sobre um UDFS instalado, bem como sobre qualquer sistema de ficheiros instalado.

**Instalar um UDFS do sistema de ficheiros integrado:** Os comandos Adicionar Sistema de Ficheiros Instalado (ADDMFS) e MOUNT tornam os objectos de um sistema de ficheiros acessíveis ao espaço de nome do sistema de ficheiros integrado. Para instalar um UDFS, tem de especificar \*UDFS para o parâmetro TYPE no comando ADDMFS.

**Nota:** Não pode instalar um UDFS por cima de outro num ASP independente.

**Desinstalar um UDFS do sistema de ficheiros integrado:** O comando desinstalar torna o conteúdo de um UDFS inacessível para as interfaces do sistema de ficheiros integrado. Uma vez desinstalado o UDFS, os objectos num UDFS deixarão de estar individualmente acessíveis. Os comandos Remover Sistema de Ficheiros Instalado (RMVUMFS) ou UNMOUNT tornarão um sistema de ficheiros instalado inacessível para o espaço de nome do sistema de ficheiros integrado. Se algum dos objectos do sistema de ficheiros estiver a ser utilizado (por exemplo, se um ficheiro estiver aberto) quando o comando estiver a ser executado, será apresentada uma mensagem de erro. O UDFS permanecerá instalado. Se tiver feito a instalação sobre alguma parte do UDFS, este último UDFS não poderá ser desinstalado até ser "descoberto".

Por exemplo, o utilizador instala um UDFS /disp/qasp02/joana.udfs sobre /home/julia no espaço de nome do sistema de ficheiros integrado. Se, em seguida, instalar outro sistema de ficheiros /pubs sobre /home/julia, o conteúdo de joana.udfs ficará inacessível. Além disso, não pode desinstalar joana.udfs sem desinstalar, em primeiro lugar, o segundo sistema de ficheiros de /home/julia.

**Nota:** Não pode instalar um UDFS por cima de outro num ASP independente.

**Guardar e restaurar um UDFS do sistema de ficheiros integrado:** O utilizador pode guardar e restaurar todos os objectos do UDFS, bem como as autoridades a ele associadas. O comando Guardar (SAV) permite-lhe guardar objectos num UDFS, enquanto o comando Restaurar (RST) permite-lhe restaurar objectos do UDFS. Ambos os comandos podem ser executados, quer o UDFS esteja instalado ou desinstalado. No entanto, para guardar correctamente os atributos do UDFS e não apenas os objectos do UDFS, o UDFS deverá ser desinstalado.

**Registar em diário alterações a objectos num sistema de ficheiros UDFS:** Os objectos nos sistemas de ficheiros definidos pelo utilizador podem ser registados em diário. O principal objectivo da gestão de registos em diário consiste em permitir que o utilizador recupere as alterações ocorridas num objecto, desde a última vez que este foi guardado. Para obter mais informações sobre o registo em diário de alterações a objectos num sistema de ficheiros UDFS, consulte "Registar objectos em diário" na página 88.

**Conjuntos de Memória Auxiliar (ASPs) Independentes e UDFS:** Quando activa um ASP independente, ocorrem várias alterações dentro do sistema de ficheiros "raiz" (/):

- É criado um directório dentro do directório /dev para o ASP independente. O nome deste directório corresponde ao nome da descrição do dispositivo associado ao ASP. Se este directório já existir antes do pedido de activação, e não estiver vazio, o procedimento de activação continua, mas não será possível trabalhar com qualquer UDFS no ASP. Neste caso, desactive o ASP independente, mude o nome do directório ou remova o seu conteúdo e, em seguida, tente executar o pedido de activação novamente.

- Dentro do directório /dev/nome\_esp encontrará objectos de ficheiro especial de bloco associados a qualquer UDFS existente no ASP independente. Existirá sempre um UDFS assumido fornecido pelo sistema. O caminho para qualquer ficheiro especial de bloco do UDFS assumido é:  
/dev/nome\_esp/QDEFAULT.UDFS
- O UDFS assumido é instalado sobre o directório /nome\_esp. Não é necessário existir o directório /nome\_esp antes do pedido de activação. No entanto, se existir, tem de estar vazio. Caso não esteja vazio, o ASP continuará activo, mas o UDFS assumido não será instalado. Neste caso, mude o nome do directório ou remova o respectivo conteúdo e, em seguida, desactive e active novamente ou utilize o comando MOUNT para instalar o UDFS assumido.
- Se o ASP independente é um ASP principal ou secundário, e o UDFS assumido foi instalado com êxito, será instalado um sistema de ficheiros adicional. O sistema de ficheiros QSYS.LIB de ASP independente será instalado em /nome\_esp/QSYS.LIB. Consulte “QSYS.LIB de ASP independente” na página 40 para obter mais informações.

**Nota:** Este sistema de ficheiros não poderá ser instalado ou desinstalado, independentemente, do UDFS assumido. É sempre instalado ou desinstalado automaticamente.

## Sistema de ficheiros de biblioteca (QSYS.LIB)

O sistema de ficheiros QSYS.LIB suporta a estrutura de bibliotecas do iSeries server. Este sistema de ficheiros fornece-lhe acesso a ficheiros de base de dados e a todos os outros tipos de objectos do iSeries server que o suporte de biblioteca gere no ASP do sistema e nos ASPs base do utilizador.

Para além disso:

- Suporta todas as interfaces de utilizador e de programação que funcionam nas bibliotecas do iSeries server e nos objectos dessas bibliotecas.
- Suporta todas as linguagens de programação e utilitários que funcionam nos ficheiros de base de dados
- Fornece suporte administrativo alargado para a gestão de objectos do servidor iSeries
- Suporta operações de I/O de sequência em membros de ficheiros físicos, espaços de utilizador e ficheiros de salvaguarda.

Até à Versão 3 do OS/400, é possível que o sistema de ficheiros QSYS.LIB fosse conhecido como o sistema de ficheiros do servidor iSeries. Os programadores que utilizavam linguagens, como RPG ou COBOL, e utilitários, como DDS, para desenvolver aplicações, utilizavam o sistema de ficheiros QSYS.LIB. Os operadores de sistema que utilizavam comandos, menus e ecrãs para manipular filas de output utilizavam o sistema de ficheiros QSYS.LIB, bem como os administradores do sistema que criavam e alteravam os perfis dos utilizadores.

Todos estes utilitários e as aplicações baseadas nos mesmos funcionam tal como funcionavam antes da introdução do sistema de ficheiros integrado. No entanto, estes utilitários não podem aceder ao QSYS.LIB através da interface do sistema de ficheiros integrado.

Para obter informações sobre o QSYS.LIB, consulte Utilizar o QSYS.LIB através da interface do sistema de ficheiros integrado.

### Utilizar o QSYS.LIB através da interface do sistema de ficheiros integrado

O sistema de ficheiros QSYS.LIB pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs do utilizador e APIs do sistema de ficheiros integrado. Quando utilizar as interfaces do sistema de ficheiros integrado, deve ter em atenção as considerações e limitações que se seguem.

- “Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB” na página 38
- “Restrições de manuseamento de ficheiros no sistema de ficheiros QSYS.LIB” na página 38
- “Suporte de espaços de utilizador no sistema de ficheiros QSYS.LIB” na página 38

- “Suporte de ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB”
- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QSYS.LIB” na página 39
- “Nomes de caminho no sistema de ficheiros QSYS.LIB” na página 39
- “Ligações no sistema de ficheiros QSYS.LIB” na página 39
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB” na página 39
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB” na página 40

**Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB:** A QPWFSEVER é uma lista de autorizações (tipo de objecto \*AUTL) que fornece requisitos de acesso adicionais a todos os objectos do sistema de ficheiros QSYS.LIB que estão a ser acedidos através de clientes remotos. As autoridades especificadas nesta lista de autorizações aplicam-se a todos os objectos existentes no sistema de ficheiros QSYS.LIB.

A autoridade assumida para este objecto é PUBLIC \*USE. O administrador pode utilizar os comandos EDTAUTL (Editar Lista de Autorizações) ou WRKAUTL (Trabalhar com Listas de Autorizações) para alterar o valor desta autoridade. O administrador pode atribuir autoridade PUBLIC \*EXCLUDE à lista de autorizações, de modo a que o público em geral não possa aceder a objectos do QSYS.LIB a partir de clientes remotos.

#### **Restrições de manuseamento de ficheiros no sistema de ficheiros QSYS.LIB:**

- Não são suportados ficheiros lógicos.
- Os ficheiros físicos suportados para o acesso em modo de texto são ficheiros físicos descritos pelo programa, que contêm um único campo, bem como ficheiros físicos origem, que contêm um único campo de texto. Os ficheiros físicos suportados para o acesso em modo binário incluem ficheiros físicos descritos externamente, para além dos ficheiros suportados para acesso em modo de texto.
- O bloqueio de intervalo de bytes não é suportado. (Para obter mais informações sobre o bloqueio de intervalo de bytes, consulte o tópico fcntl() do Centro de Informações do iSeries.)
- Se qualquer trabalho tiver um membro de ficheiro de base de dados aberto, é concedido apenas a um trabalho o acesso de escrita para esse membro do ficheiro, em qualquer altura. Aos outros pedidos, só é permitido o acesso de leitura.

**Suporte de espaços de utilizador no sistema de ficheiros QSYS.LIB:** O QSYS.LIB suporta operações de input e output de sequência para objectos de espaço de utilizador. Por exemplo, um programa pode escrever dados de sequência num espaço de utilizador e ler dados a partir de um espaço de utilizador. O tamanho máximo de um espaço de utilizador é de 16 776 704 bytes.

Repare que os espaços de utilizador não estão identificados com um CCSID (identificador do conjunto de caracteres codificados). Deste modo, o CCSID devolvido é o CCSID assumido para o trabalho.

**Suporte de ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB:** O sistema de ficheiros QSYS.LIB suporta operações de I/O de sequência para objectos de ficheiros de salvaguarda. Por exemplo, um ficheiro de salvaguarda existente tem dados que podem ser lidos ou copiados para outro ficheiro, até ser necessário colocá-los num objecto diferente de ficheiro de salvaguarda existente e vazio. Quando um ficheiro de salvaguarda é aberto para escrita, não são permitidas outras ocorrências abertas do ficheiro. Um ficheiro de salvaguarda **permite** várias ocorrências abertas para leitura, desde que nenhum trabalho tenha mais do que uma ocorrência aberta do ficheiro para leitura. Não é possível abrir um ficheiro de salvaguarda para acesso de leitura/escrita. As operações de I/O de sequência em dados do ficheiro de salvaguarda não são permitidas quando estão a ser executados vários módulos num trabalho.

As operações de I/O de sequência num ficheiro de salvaguarda não são suportadas quando o referido ficheiro ou o respectivo directório estão a ser exportados através do servidor Sistema de Ficheiros da Rede. É, no entanto, possível ter acesso às mesmas a partir de clientes de PC e através do sistema de ficheiros QFileSvr.400.

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QSYS.LIB:** Normalmente, o sistema de ficheiros QSYS.LIB não distingue as maiúsculas e as minúsculas dos nomes dos objectos. A procura dos nomes dos objectos tem o mesmo resultado, independentemente de os caracteres dos nomes estarem em maiúsculas ou minúsculas.

No entanto, se um nome estiver entre plicas, o tipo de letra dos caracteres mantém-se inalterado. Por conseguinte, a procura que envolva nomes escritos entre plicas é sensível a maiúsculas e minúsculas do nome entre plicas.

#### **Nomes de caminho no sistema de ficheiros QSYS.LIB:**

- Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto. Por exemplo:

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

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

O nome do objecto e o tipo de objecto são separados por um ponto (.). Os objectos numa biblioteca podem ter o mesmo nome se tiverem tipos de objectos diferentes, pelo que o tipo de objecto tem de ser especificado para identificar o objecto de forma exclusiva.

- O nome do objecto em cada componente pode ter até 10 caracteres de comprimento e o tipo de objecto pode ter até 6 caracteres de comprimento.
- A hierarquia de directórios do QSYS.LIB pode ter dois ou três níveis (dois ou três componentes no nome de caminho), dependendo do tipo de objecto que está a ser acedido. Se o objecto for um ficheiro de base de dados, a hierarquia pode conter três níveis (biblioteca, ficheiro, membro); caso contrário, só poderão existir dois níveis (biblioteca, objecto). A combinação do comprimento de cada nome do componente e o número de níveis de directórios determina o comprimento máximo do nome de caminho.

Se os sistemas de ficheiros "raiz" (/) e QSYS.LIB estiverem incluídos como os dois primeiros níveis, a hierarquia de directórios para QSYS.LIB pode ter até cinco níveis de profundidade.

- Os caracteres nos nomes são convertidos no CCSID 37, quando os nomes são armazenados. No entanto, os nomes entre plicas são armazenados utilizando o CCSID do trabalho.

Para obter mais informações sobre o CCSID, consulte o tópico Globalization do Centro de Informações do iSeries.

**Ligações no sistema de ficheiros QSYS.LIB:** Não é possível criar ou armazenar ligações simbólicas no sistema de ficheiros QSYS.LIB.

A relação entre uma biblioteca e os objectos numa biblioteca é equivalente a uma ligação permanente entre a biblioteca e cada objecto na biblioteca. O sistema de ficheiros integrado considera a relação biblioteca-objecto como uma ligação. Assim, é possível ligar um sistema de ficheiros que suporte ligações simbólicas a um objecto do sistema de ficheiros QSYS.LIB.

Consulte "Ligação" na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB:** Os comandos incluídos em "Aceder através de comandos de CL" na página 62 podem funcionar no sistema de ficheiros QSYS.LIB, com as seguintes excepções:

- O comando ADDLNK só pode ser utilizado para criar uma ligação simbólica a um objecto no QSYS.LIB.
- As operações em ficheiros só podem ser efectuadas nos ficheiros físicos descritos pelo programa e nos ficheiros físicos origem.
- Os comandos STRJRN e ENDJRN não podem ser utilizados em ficheiros físicos de base de dados.

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em "Aceder através de menus e ecrãs" na página 61.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB:** As APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros QSYS.LIB, com as seguintes excepções:

- As operações em ficheiros só podem ser efectuadas nos ficheiros físicos descritos pelo programa e nos ficheiros físicos origem.
- A função symlink() só pode ser utilizada para ligar a um objecto do QSYS.LIB a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- As APIs QjoStartJournal() e QjoEndJournal() não podem ser utilizadas em ficheiros físicos de base de dados.

## **QSYS.LIB de ASP independente**

O sistema de ficheiros QSYS.LIB de ASP independente suporta a estrutura de biblioteca iSeries server em conjuntos de memória auxiliar (ASPs) independentes criados e definidos por si. Este sistema de ficheiros fornece acesso a ficheiros de base de dados e a todos os outros tipos de objectos do iSeries server que o suporte de biblioteca gere nos ASPs independentes.

Para além disso:

- Suporta todas as interfaces de utilizador e de programação que funcionam nas bibliotecas do iSeries server e nos objectos dessas bibliotecas em ASPs independentes
- Suporta todas as linguagens de programação e utilitários que funcionam nos ficheiros de base de dados
- Fornece suporte administrativo alargado para gerir objectos do iSeries server
- Suporta operações de I/O de sequência em membros de ficheiros físicos, espaços de utilizador e ficheiros de salvaguarda.

Para obter mais informações sobre o sistema de ficheiros QSYS.LIB de ASP independente, consulte Utilizar QSYS.LIB de ASP independente através da interface do sistema de ficheiros integrado.

## **Utilizar QSYS.LIB de ASP independente através da interface do sistema de ficheiros integrado**

O sistema de ficheiros QSYS.LIB de ASP independente pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs de utilizador e APIs do sistema de ficheiros integrado. Quando utilizar as interfaces do sistema de ficheiros integrado, deve ter em atenção as considerações e limitações que se seguem.

- “Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB de ASP independente” na página 41
- “Restrições de manuseamento de ficheiros no sistema de ficheiros QSYS.LIB de ASP independente” na página 41
- “Suporte para espaços de utilizador no sistema de ficheiros QSYS.LIB de ASP independente” na página 41
- “Suporte para ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB de ASP independente” na página 41
- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QSYS.LIB de ASP independente” na página 41
- “Nomes de caminho no sistema de ficheiros QSYS.LIB de ASP independente” na página 42
- “Ligações no sistema de ficheiros QSYS.LIB de ASP independente” na página 42
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB de ASP independente” na página 42
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB de ASP independente” na página 43

**Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB de ASP independente:** A QPWFSEVER é uma lista de autorizações (tipo de objecto \*AUTL) que fornece requisitos de acesso adicionais para todos os objectos do sistema de ficheiros QSYS.LIB de ASP independente que sejam acedidos através de clientes remotos. As autoridades especificadas nesta lista de autorizações aplicam-se a todos os objectos no sistema de ficheiros QSYS.LIB de ASP independente.

A autoridade assumida para este objecto é PUBLIC \*USE. O administrador pode utilizar os comandos EDTAUTL (Editar Lista de Autorizações) ou WRKAUTL (Trabalhar com Listas de Autorizações) para alterar o valor desta autoridade. O administrador pode atribuir autoridade PUBLIC \*EXCLUDE à lista de autorizações, de modo a que o público em geral não possa aceder a objectos do QSYS.LIB de ASP independente a partir de clientes remotos.

**Restrições de manuseamento de ficheiros no sistema de ficheiros QSYS.LIB de ASP independente:**

- Não são suportados ficheiros lógicos.
- Os ficheiros físicos suportados para o acesso em modo de texto são ficheiros físicos descritos pelo programa, que contêm um único campo, bem como ficheiros físicos origem, que contêm um único campo de texto. Os ficheiros físicos suportados para o acesso em modo binário incluem ficheiros físicos descritos externamente, para além dos ficheiros suportados para acesso em modo de texto.
- O bloqueio de intervalo de bytes não é suportado. (Para obter mais informações sobre o bloqueio de intervalo de bytes, consulte o tópico `fcntl()` do Centro de Informações do iSeries.)
- Se qualquer trabalho tiver um membro de ficheiro de base de dados aberto, é concedido apenas a um trabalho o acesso de escrita para esse membro do ficheiro, em qualquer altura. Aos outros pedidos, só é permitido o acesso de leitura.

**Suporte para espaços de utilizador no sistema de ficheiros QSYS.LIB de ASP independente:** O QSYS.LIB de ASP independente suporta operações de input e output de sequência para objectos de espaço de utilizador. Por exemplo, um programa pode escrever dados de sequência num espaço de utilizador e ler dados a partir de um espaço de utilizador. O tamanho máximo de um espaço de utilizador é de 16 776 704 bytes.

Repare que os espaços de utilizador não estão identificados com um CCSID (identificador do conjunto de caracteres codificados). Deste modo, o CCSID devolvido é o CCSID assumido para o trabalho.

**Suporte para ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB de ASP independente:** O sistema de ficheiros QSYS.LIB de ASP independente suporta operações I/O de sequência para objectos de ficheiros de salvaguarda. Por exemplo, um ficheiro de salvaguarda existente tem dados que podem ser lidos ou copiados para outro ficheiro, até ser necessário colocá-los num objecto diferente de ficheiro de salvaguarda existente e vazio. Quando um ficheiro de salvaguarda é aberto para escrita, não são permitidas outras ocorrências abertas do ficheiro. Um ficheiro de salvaguarda **permite** várias ocorrências abertas para leitura, desde que nenhum trabalho tenha mais do que uma ocorrência aberta do ficheiro para leitura. Não é possível abrir um ficheiro de salvaguarda para acesso de leitura/escrita. As operações de I/O de sequência em dados do ficheiro de salvaguarda não são permitidas quando estão a ser executados vários módulos num trabalho.

As operações de I/O de sequência num ficheiro de salvaguarda não são suportadas quando o referido ficheiro ou o respectivo directório estão a ser exportados através do servidor Sistema de Ficheiros da Rede. É, no entanto, possível ter acesso às mesmas a partir de clientes de PC e através do sistema de ficheiros QFileSvr.400.

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QSYS.LIB de ASP independente:** Normalmente, o sistema de ficheiros QSYS.LIB de ASP independente não distingue as maiúsculas e as minúsculas nos nomes de objectos. A procura dos nomes dos objectos tem o mesmo resultado, independentemente de os caracteres dos nomes estarem em maiúsculas ou minúsculas.

No entanto, se um nome estiver entre plicas, o tipo de letra dos caracteres mantém-se inalterado. Por conseguinte, a procura que envolva nomes escritos entre plicas é sensível a maiúsculas e minúsculas do nome entre plicas.

#### **Nomes de caminho no sistema de ficheiros QSYS.LIB de ASP independente:**

- Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto. Por exemplo:

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

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

em que nome\_asp é o nome do ASP independente. O nome do objecto e o tipo de objecto são separados por um ponto (.). Os objectos numa biblioteca podem ter o mesmo nome se tiverem tipos de objectos diferentes, pelo que o tipo de objecto tem de ser especificado para identificar o objecto de forma exclusiva.

- O nome do objecto em cada componente pode ter até 10 caracteres de comprimento e o tipo de objecto pode ter até 6 caracteres de comprimento.
- A hierarquia de directórios do QSYS.LIB de ASP independente pode ter dois ou três níveis de profundidade (dois ou três componentes no nome de caminho), dependendo do tipo de objecto que está a ser acedido. Se o objecto for um ficheiro de base de dados, a hierarquia pode conter três níveis (biblioteca, ficheiro, membro); caso contrário, só poderão existir dois níveis (biblioteca, objecto). A combinação do comprimento de cada nome do componente e o número de níveis de directórios determina o comprimento máximo do nome de caminho.

Se forem incluídos/, nome\_asp e QSYS.LIB como os primeiros três níveis, a hierarquia de directórios para o sistema de ficheiros QSYS.LIB de ASP independente pode ter até seis níveis de profundidade.

- Os caracteres nos nomes são convertidos no CCSID 37, quando os nomes são armazenados. No entanto, os nomes entre plicas são armazenados utilizando o CCSID do trabalho.

Para obter mais informações sobre o CCSID, consulte o tópico Globalization do Centro de Informações do iSeries.

**Ligações no sistema de ficheiros QSYS.LIB de ASP independente:** Não é possível criar nem armazenar ligações simbólicas no sistema de ficheiros QSYS.LIB de ASP independente.

A relação entre uma biblioteca e os objectos numa biblioteca é equivalente a uma ligação permanente entre a biblioteca e cada objecto na biblioteca. O sistema de ficheiros integrado considera a relação biblioteca-objecto como uma ligação. Assim, é possível ligar um sistema de ficheiros que suporte ligações simbólicas a um objecto do sistema de ficheiros QSYS.LIB de ASP independente.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB de ASP independente:** Os comandos incluídos em “Aceder através de comandos de CL” na página 62 podem funcionar no sistema de ficheiros QSYS.LIB de ASP independente, excepto quanto ao seguinte:

- O comando ADDLNK só pode ser utilizado para criar uma ligação simbólica a um objecto do QSYS.LIB de ASP independente.
- As operações em ficheiros só podem ser efectuadas nos ficheiros físicos descritos pelo programa e nos ficheiros físicos origem.
- Os comandos STRJRN e ENDJRN não podem ser utilizados em ficheiros físicos de base de dados.
- Não pode mover bibliotecas do sistema de ficheiros QSYS.LIB de ASP independente para conjuntos de memória auxiliar (ASPs) base utilizando o comando MOV. No entanto, pode mover bibliotecas do QSYS.LIB de ASP independente para o ASP do sistema ou outros ASPs independentes.
- Se está a utilizar SAV ou RST para guardar ou restaurar objectos de biblioteca num ASP independente, esse ASP independente tem de estar associado ao trabalho que executa a operação SAV ou RST ou o

ASP independente tem de ser especificado no parâmetro ASPDEV. A convenção de nomenclatura de nomes de caminho/nome\_asp/QSYS.LIB/tipo.objecto não é suportada em SAV e RST.

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder através de menus e ecrãs” na página 61.

### **Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB de ASP**

**independente:** As APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros QSYS.LIB de ASP independente, com as seguintes excepções:

- As operações em ficheiros só podem ser efectuadas nos ficheiros físicos descritos pelo programa e nos ficheiros físicos origem.
- A função symlink() só pode ser utilizada para ligar a um objecto do QSYS.LIB de ASP independente a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- As APIs QjoStartJournal() e QjoEndJournal() não podem ser utilizadas em ficheiros físicos de base de dados.

## **Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)**

O sistema de ficheiros QDLS suporta a estrutura de pastas. Faculta o acesso a documentos e pastas.

Para além disso:

- Suporta pastas do servidor iSeries e objectos da biblioteca de documentos (DLOs).
- Suporta dados armazenados em ficheiros de dados contínuos.

Para obter mais informações sobre o QDLS, consulte Utilizar o QDLS através da interface do sistema de ficheiros integrado.

### **Utilizar o QDLS através da interface do sistema de ficheiros integrado**

O sistema de ficheiros QDLS pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs do utilizador e APIs do sistema de ficheiros integrado. Quando utilizar as interfaces do sistema de ficheiros integrado, deve ter em atenção as considerações e limitações que se seguem.

Para obter mais informações sobre o sistema de ficheiros QDLS, consulte os seguintes tópicos.

- “Sistema de ficheiros integrado e HFS no sistema de ficheiros QDLS”
- “Inscrição de utilizadores no sistema de ficheiros QDLS” na página 44
- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QDLS” na página 44
- “Nomes de caminhos no sistema de ficheiros QDLS” na página 44
- “Ligações no sistema de ficheiros QDLS” na página 44
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QDLS” na página 44
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QDLS” na página 45

**Sistema de ficheiros integrado e HFS no sistema de ficheiros QDLS:** As operações podem ser executadas em objectos no sistema de ficheiros QDLS, não apenas através de comandos de CL de Objectos da Biblioteca de Documentos (DLO), mas também através da interface do sistema de ficheiros integrado ou das APIs fornecidas por um sistema de ficheiros hierárquico, conhecido por HFS. Enquanto o sistema de ficheiros integrado se baseia no modelo do programa do Integrated Language Environment (ILE), o HFS baseia-se no modelo de programa do servidor iSeries original.

As APIs do HFS permitem-lhe executar algumas operações adicionais que o sistema de ficheiros integrado não suporta. Nomeadamente, pode utilizar as APIs de HFS para ter acesso e alterar os atributos expandidos de directório (também conhecidos por *atributos de entrada de directório*). Tenha em

atenção que as regras de nomenclatura para a utilização de APIs de HFS são diferentes das regras de nomenclatura para as APIs que utilizem a interface do sistema de ficheiros integrado.

Para obter mais informações sobre o HFS, consulte o tópico Hierarchical File System APIs do iSeries Information Center.

**Inscrição de utilizadores no sistema de ficheiros QDLS:** Tem de estar inscrito no directório de distribuição do sistema para trabalhar com objectos do QDLS.

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QDLS:** O QDLS converte os caracteres ingleses alfabéticos em minúsculas de a a z em maiúsculas, quando utilizados em nomes de objectos. Por conseguinte, a procura de nomes dos objectos utilizando apenas esses caracteres não é sensível a maiúsculas e minúsculas.

No QDLS, todos os outros caracteres são sensíveis a maiúsculas e minúsculas.

Para obter mais detalhes, consulte o tópico Folder and Document Name do iSeries Information Center.

**Nomes de caminhos no sistema de ficheiros QDLS:**

- Cada componente do nome de caminho pode ser constituído por apenas um nome, tal como:

`/QDLS/ARQ1/DOC1`

ou por um nome e uma extensão (semelhante a uma extensão de ficheiro do DOS), tal como:

`/QDLS/ARQ1/DOC1.TXT`

- O nome em cada componente pode ter até 8 caracteres de comprimento e a extensão (se existir) pode ter até 3 caracteres de comprimento. O comprimento máximo do nome de caminho é 82 caracteres, assumindo um nome de caminho absoluto começado por /QDLS.
- A hierarquia de directórios do QDLS pode ter 32 níveis de profundidade. Se / e QDLS forem incluídos como os primeiros dois níveis, a hierarquia de directórios poderá ter 34 níveis.
- Os caracteres dos nomes são convertidos na página de códigos do trabalho quando os nomes são armazenados, a menos que tenha sido criada a área Q0DEC500 na biblioteca QUSRSYS. Se esta área de dados existir, os caracteres dos nomes são convertidos na página de códigos 500 quando os nomes forem armazenados. Esta função facultava compatibilidade com o comportamento do sistema de ficheiros QDLS em edições anteriores. Um nome pode ser rejeitado se não puder ser convertido na página de códigos adequada.

Para obter mais informações sobre páginas de códigos, consulte o tópico Globalization do iSeries Information Center.

**Ligações no sistema de ficheiros QDLS:** Não é possível criar nem armazenar ligações simbólicas no sistema de ficheiros QDLS.

O sistema de ficheiros integrado considera a relação entre uma pasta e os objectos da biblioteca de documentos como uma ligação entre a pasta e cada objecto na pasta. Assim, é possível estabelecer ligação a um objecto do sistema de ficheiros QDLS, a partir de um sistema de ficheiros que suporte ligações simbólicas.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QDLS:** Os comandos incluídos em “Aceder através de comandos de CL” na página 62 podem funcionar no sistema de ficheiros QDLS, com as seguintes excepções:

- O comando ADDLNK só pode ser utilizado para ligar a um objecto do QDLS a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- Os comandos CHKIN e CHKOUT são suportados para ficheiros, mas não para directórios.

- Os comandos APYJRNCHG, CHGJRNOBJ, ENDJRN, SNDJRNE, e STRJRN não são suportados.

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder através de menus e ecrãs” na página 61.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QDLS:** As APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros QDLS, com as seguintes excepções:

- A função symlink() só pode ser utilizada para ligar a um objecto do QDLS a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- Não são suportadas as seguintes funções:

```
givedescriptor()
ioctl()
link()
QjoEndJournal()
QjoRetrieveJournalInformation()
QJORJIDI()
QJOSJRNE()
QjoStartJournal()
Qp0lGetPathFromFileID()
readlink()
takedescriptor()
```

## Sistema de Ficheiros Óptico (QOPT)

O sistema de ficheiros QOPT permite o acesso a dados de sequência armazenados em suportes ópticos.

Para além disso:

- Fornece uma estrutura de directórios hierárquica, semelhante à dos sistemas operativos de PC, tal como DOS e OS/2.
- Está optimizado para input e output de ficheiros de dados contínuos.
- Suporta dados armazenados em ficheiros de dados contínuos.

Para obter mais informações sobre o QOPT, consulte Utilizar o QOPT através do sistema de ficheiros integrado.

### Utilizar o QOPT através do sistema de ficheiros integrado

O sistema de ficheiros QOPT pode ser acedido através do sistema de ficheiros integrado, utilizando o servidor de PC ou comandos, ecrãs de utilizador e APIs do sistema de ficheiros integrado. Quando utilizar a interface do sistema de ficheiros integrado, deve ter em atenção as considerações e limitações que se seguem.

Para obter mais informações sobre o sistema de ficheiros QOPT, consulte os seguintes tópicos.

- “Sistema de ficheiros integrado e HFS no sistema de ficheiros QOPT” na página 46
- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QOPT” na página 46
- “Nomes de caminhos no sistema de ficheiros QOPT” na página 46
- “Ligações no sistema de ficheiros QOPT” na página 46
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOPT” na página 46
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOPT” na página 47

Para mais detalhes, consulte a publicação Optical Support  .

**Sistema de ficheiros integrado e HFS no sistema de ficheiros QOPT:** No sistema de ficheiros QOPT, as operações podem ser executadas em objectos, através da interface do sistema de ficheiros integrado ou das APIs fornecidas por um sistema de ficheiros hierárquico conhecido por HFS. Enquanto o sistema de ficheiros integrado se baseia no modelo do programa do Integrated Language Environment (ILE), o HFS baseia-se no modelo de programa do servidor iSeries original.

As APIs do HFS permitem-lhe executar algumas operações adicionais que o sistema de ficheiros integrado não suporta. Nomeadamente, pode utilizar as APIs do HFS para ter acesso e alterar atributos expandidos de directório (também conhecidos por *atributos de entrada de directório*) ou para trabalhar com ficheiros ópticos retidos. Tenha em atenção que as regras de nomenclatura para a utilização de APIs de HFS são diferentes das regras de nomenclatura para as APIs que utilizem a interface do sistema de ficheiros integrado.

Para obter mais informações sobre APIs de HFS, consulte o tópico Hierarchical File System APIs do iSeries Information Center ou a publicação Optical Support  .

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QOPT:** Dependendo do formato do suporte óptico, as maiúsculas ou minúsculas podem ou não ser mantidas durante a criação de ficheiros ou directórios do QOPT. Contudo, as pesquisas em ficheiros e directórios não são sensíveis a maiúsculas e minúsculas, independentemente do formato do suporte óptico.

#### **Nomes de caminhos no sistema de ficheiros QOPT:**

- O nome de caminho tem de começar por uma barra (/). O caminho é constituído pelo nome do sistema de ficheiros, pelo nome do volume e pelos nomes de directório e subdirectórios e nome do ficheiro. Por exemplo:  
`/QOPT/NOMEVOLUME/NOMEDIRECTÓRIO/NOMESUBDIRECTÓRIO/NOMEFICH`
- É necessário o nome do sistema de ficheiros QOPT.
- O volume e o comprimento do nome do caminho variam conforme o formato do suporte óptico.
- Pode especificar /QOPT no nome do caminho ou incluir um ou mais directórios ou subdirectórios no nome de caminho. Os nomes de directórios e de ficheiros permitem a utilização de qualquer carácter, excepto X'00' a X'3F', X'FF'. Podem ser aplicáveis restrições adicionais com base no formato do suporte óptico.
- O nome do ficheiro é o último elemento no nome de caminho. O comprimento do nome do ficheiro é limitado pelo comprimento do nome do directório no caminho.

Para obter mais detalhes sobre regras de nomes de caminho no sistema de ficheiros QOPT, consulte a secção “ Path Name Rules” na publicação Optical Support  .

**Ligações no sistema de ficheiros QOPT:** O sistema de ficheiros QOPT suporta apenas uma ligação a um objecto. Não é possível criar nem armazenar ligações simbólicas no QOPT. No entanto, os ficheiros no QOPT podem ser acedidos utilizando uma ligação simbólica do sistema de ficheiros “raiz” (/) ou QOpenSys.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOPT:** A maior parte dos comandos incluídos em “Aceder através de comandos de CL” na página 62 pode funcionar no sistema de ficheiros QOPT. Existem, contudo, algumas excepções no sistema de ficheiros QOPT. Não se esqueça que pode não ser seguro utilizar estes comandos de CL num processo com capacidade para

vários módulos; podem ser aplicáveis determinadas restrições, dependendo do formato do suporte óptico. Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder através de menus e ecrãs” na página 61.

Não são suportados pelo sistema de ficheiros QOPT os seguintes comandos do sistema de ficheiros integrado:

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

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOPT:** Todas as APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros “raiz” (/) com protecção para módulos, à excepção de:

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

## Sistema de ficheiros NetWare (QNetWare)

O sistema de ficheiros QNetWare faculta o acesso a dados num Integrated xSeries Server for iSeries local ou remoto que utilize o Novell NetWare 4.10 ou 4.11 ou a Servidores de PC autónomos que utilizem o Novell NetWare 3.12, 4.10, 4.11 ou 5.0.

Para além disso:

- Permite o acesso a objectos de Serviços de Directório de NetWare (NDS).
- Suporta dados armazenados em ficheiros de dados contínuos.
- Permite a instalação dinâmica de Sistemas de Ficheiros Netware no espaço de nome local

**Nota:** O sistema de ficheiros QNetWare só está disponível quando o programa NetWare Enhanced Integration for iSeries 400<sup>®</sup>, opção 25 do BOSS, está instalado no sistema. Depois do IPL que se segue à instalação, o directório /QNetWare e os respectivos subdirectórios aparecem como parte da estrutura de directórios do sistema de ficheiros integrado.

Consulte os tópicos que se seguem para obter mais informações sobre o sistema de ficheiros QNetWare:

- Instalar sistemas de ficheiros NetWare
- Estrutura de directórios QNetWare
- Utilizar QNetWare através da interface do sistema de ficheiros integrado

## Instalar sistemas de ficheiros NetWare

Os sistemas de ficheiros NetWare localizados nos servidores Novell NetWare podem ser instalados nos sistemas de ficheiros “raiz” (/), QOpenSys ou noutros sistemas de ficheiros, de modo a facilitar o acesso e obter um melhor funcionamento que no directório /QNetWare. A instalação de sistemas de ficheiros

Netware também pode ser utilizada para tirar partido das opções do comando Adicionar Sistema de Ficheiros Instalado (ADDMFS), tal como instalar um sistema de ficheiros de leitura e escrita como sendo só de leitura.

Os sistemas de ficheiros NetWare podem ser instalados utilizando um caminho de NDS ou especificando um caminho de NetWare no formato `SERVIDOR/VOLUME:directório/directório`. Por exemplo, para instalar o directório passagem localizado no volume Ninho, no servidor Dreyfuss, deve utilizar a seguinte sintaxe:

```
DREYFUSS/NINHO:passagem
```

Esta sintaxe de caminho é muito semelhante à sintaxe dos comandos MAP de NetWare. Os caminhos NDS podem ser utilizados para especificar um caminho para o volume NetWare, mas não podem ser, eles próprios, instalados.

## Estrutura de directórios QNetWare

A estrutura de directórios /QNetWare representa múltiplos sistemas de ficheiros distintos:

- A estrutura representa servidores e volumes Novell NetWare existentes na rede, com o seguinte formato:

```
/QNetWare/SERVIDOR.SVR/VOLUME
```

A extensão `.SVR` é utilizada para representar um servidor Novell NetWare.

- Quando um volume de um servidor é acedido através de menus, comandos ou APIs do sistema de ficheiros integrado, o directório raiz do volume NetWare é automaticamente instalado no directório VOLUME em /QNetWare.
- QNetWare representa árvores NDS na rede com o seguinte formato:

```
/QNetWare/CORP_TREE.TRE/USA.C/ORG.0/ORG_UNIT.OU/SVR1_VOL.CN
```

A extensão `.TRE` é utilizada para representar árvores NDS, `.C` representa os países, `.0` representa organizações, `.OU` representa unidades organizacionais e `.CN` é utilizado para representar nomes comuns. Se um volume da Novell Netware for acedido pelo caminho NDS através de um objecto volume ou de um nome alternativo de um objecto volume, o directório raiz também será automaticamente instalado no objecto NDS.

## Utilizar QNetWare através da interface do sistema de ficheiros integrado

O sistema de ficheiros QNetWare pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros OS/400 ou comandos, ecrãs de utilizador e APIs do sistema de ficheiros integrado. Deve ter em atenção as considerações, limitações e dependências que se seguem.

Para ver mais informações sobre o sistema de ficheiros QNetWare, consulte:

- “Autoridades e propriedade no sistema de ficheiros QNetWare”
- “Auditoria no sistema de ficheiros QNetWare” na página 49
- “Ficheiros e directórios no sistema de ficheiros QNetWare” na página 49
- “Objectos NDS no sistema de ficheiros QNetWare” na página 49
- “Ligações no sistema de ficheiros QNetWare” na página 49
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNetWare” na página 49
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNetWare” na página 49

**Autoridades e propriedade no sistema de ficheiros QNetWare:** Os ficheiros e os directórios do QNetWare são armazenados e geridos pelos servidores Novell NetWare. Quando utilizar comandos e APIs para obter ou definir as autoridades de proprietários ou utilizadores, QNetWare definirá a correspondência de utilizadores de NetWare como utilizadores do servidor iSeries, com base no nome de um utilizador. Se o nome de NetWare exceder dez caracteres ou se um utilizador do servidor iSeries correspondente não existir, a autoridade não será definida. Os proprietários que não puderem ser definidos serão automaticamente definidos com o perfil do utilizador QDFTOWN. As autoridades dos

utilizadores podem ser visualizadas e alteradas utilizando os comandos WRKAUT e CHGAUT. Quando são transferidas de e para o servidor, as autoridades são correlacionadas com as autoridades do servidor iSeries.

**Auditoria no sistema de ficheiros QNetWare:** Apesar da Novell NetWare suportar a auditoria de ficheiros e de directórios, o sistema de ficheiros QNetWare não pode alterar os valores auditados desses objectos. Por conseguinte, o comando CHGAUD não é suportado.

**Ficheiros e directórios no sistema de ficheiros QNetWare:** O sistema de ficheiros QNetWare não mantém as maiúsculas ou minúsculas utilizadas quando os ficheiros e directórios são introduzidos num comando ou numa API. Todos os nomes são definidos para maiúsculas na transmissão para o servidor NetWare. A Novell NetWare também suporta os espaços de nome de múltiplas plataformas como, por exemplo DOS, OS/2, Apple Macintosh e NFS. O sistema de ficheiros QNetWare suporta apenas o espaço de nome do DOS. Uma vez que o espaço de nome do DOS é necessário em todos os volumes Novell NetWare, todos os ficheiros e directórios aparecerão no sistema de ficheiros QNetWare.

**Objectos NDS no sistema de ficheiros QNetWare:** O sistema de ficheiros QNetWare suporta a apresentação de nomes NDS em maiúsculas e minúsculas.

**Ligações no sistema de ficheiros QNetWare:** O sistema de ficheiros QNetWare suporta apenas uma ligação a um objecto. Não é possível criar nem armazenar ligações simbólicas no QNetWare. No entanto, podem ser criadas ligações simbólicas nos directórios "raiz" (/) ou QOpenSys que apontem para um ficheiro ou directório QNetWare.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNetWare:** Os comandos incluídos em "Aceder através de comandos de CL" na página 62 podem funcionar no sistema de ficheiros QNetWare, com as seguintes excepções:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGJRNOBJ
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Para além dos comandos anteriores, não é possível utilizar os comandos que se seguem em objectos, servidores ou volumes NDS:

- CHGOWN
- CPYFRMSTMF
- CPYTOSTMF
- CRTDIR

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNetWare:** As APIs incluídas em "Executar operações utilizando APIs" na página 98 podem funcionar no sistema de ficheiros QNetWare, à excepção das seguintes APIs:

- givedescriptor()
- link()

QjoEndJournal()  
QjoRetrieveJournalInformation()  
QJORJIDI()  
QJOSJRNE()  
QjoStartJournal()  
readlink()  
symlink()  
takedescriptor()

Para além das APIs anteriores, não podem ser utilizadas as APIs que se seguem em objectos, servidores ou volumes NDS:

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

## **Sistema de ficheiros do iSeries NetClient (QNTC)**

O sistema de ficheiros QNTC permite o acesso a dados e objectos armazenados num Integrated xSeries Server for iSeries local ou remoto que utilize o Windows NT 4.0 Server ou posterior, ou num servidor autónomo. Permite às aplicações do servidor iSeries utilizarem os mesmos dados que os clientes do Windows.

O sistema de ficheiros QNTC faz parte do sistema operativo OS/400 base. Não é necessário ter instalado o programa iSeries 400 Integration with Windows NT Server, opção 29 do Sistema Operativo, para aceder ao /QNTC.

Para obter mais informações sobre o QNTC, consulte Utilizar o QNTC através da interface do sistema de ficheiros integrado.

### **Utilizar o QNTC através da interface do sistema de ficheiros integrado**

Utilizando o iSeries NetServer, iSeries Navigator, os comandos, ecrãs de utilizador ou APIs do sistema de ficheiros integrado, pode aceder ao sistema de ficheiros QNTC através da interface do sistema de ficheiros integrado. Deve ter em atenção as considerações e limitações que se seguem.

Os tópicos que se seguem fornecem mais informações sobre o sistema de ficheiros QNTC.

- “Autoridades e propriedade no sistema de ficheiros QNTC” na página 51
- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QNTC” na página 51
- “Nomes de caminhos no sistema de ficheiros QNTC” na página 51
- “Ligações no sistema de ficheiros QNTC” na página 51

- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNTC”
- “Utilizar o comando MKDIR no sistema de ficheiros QNTC” na página 52
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNTC” na página 52
- “Activar Kerberos para utilização com o sistema de ficheiros QNTC” na página 53

**Autoridades e propriedade no sistema de ficheiros QNTC:** O sistema de ficheiros QNTC não suporta o conceito de propriedade de um ficheiro ou directório. As tentativas para utilizar um comando ou uma API para alterar a propriedade de ficheiros armazenados no QNTC não terão êxito. Um perfil de utilizador do sistema, denominado QDFTOWN, é o proprietário de todos os ficheiros e directórios no QNTC.

A autoridade para ficheiros e directórios do servidor de NT é administrada a partir do servidor do Windows NT. O QNTC não suporta os comandos WRKAUT e CHGAUT.

**Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QNTC:** O sistema de ficheiros QNTC preserva o mesmo formato de maiúsculas e minúsculas em que os nomes de objectos são introduzidos, mas não distingue maiúsculas e minúsculas nos nomes. A procura dos nomes dos objectos tem o mesmo resultado, independentemente de os caracteres dos nomes estarem em maiúsculas ou minúsculas.

#### **Nomes de caminhos no sistema de ficheiros QNTC:**

- O nome de caminho tem de começar por uma barra e pode conter até 255 caracteres.
- Os nomes de caminho são sensíveis a maiúsculas e minúsculas.
- O caminho consiste no nome do sistema de ficheiros, no nome do servidor do Windows NT, no nome da partilha, nos nomes de directórios e subdirectórios e no nome do objecto. Os nomes de caminho têm o seguinte formato:  
     /QNTC/Nomeservidor/Nomepartilha/Directório/ . . . /objecto  
     (QNTC é uma parte obrigatória do nome de caminho.)
- O nome do servidor pode ter até 15 caracteres. Tem de fazer parte do caminho.
- O nome da partilha pode ter até 12 caracteres.
- Cada componente do nome de caminho, após o nome da partilha, pode ter até 255 caracteres.
- Geralmente, no QNTC, estão disponíveis 130 níveis de hierarquia. Se todos os componentes do nome de caminho estiverem incluídos como níveis de hierarquia, a hierarquia de directórios pode ter até 132 níveis de profundidade.
- Os nomes são armazenados em CCSID Unicode.
- Cada servidor funcional do Windows NT na sub-rede local aparece automaticamente como um directório em /QNTC. Utilize o comando Criar Directório (MKDIR) (consulte a Tabela 7 na página 62) ou a API mkdir() (consulte “Aceder utilizando APIs” na página 80) para adicionar servidores baseados no Windows exteriores à sub-rede local.

**Ligações no sistema de ficheiros QNTC:** O sistema de ficheiros QNTC suporta apenas uma ligação a um objecto. Não pode criar ou armazenar ligações simbólicas no QNTC. Pode utilizar uma ligação simbólica a partir do sistema de ficheiros “raiz” (/) ou QOpenSys para aceder a dados do QNTC.

Consulte “Ligação” na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNTC:** Os comandos incluídos em “Aceder através de comandos de CL” na página 62 podem funcionar no sistema de ficheiros QNTC, com as seguintes excepções:

```
ADDLNK
APYJRNCHG
CHGJRNOBJ
CHGOWN
```

CHGAUT  
CHGPGP  
CHKIN  
CHKOUT  
DSPAUT  
ENDJRN  
RST (disponível nos Integrated xSeries® Servers)  
SAV (disponível nos Integrated xSeries Servers)  
SNDJRNE  
STRJRN  
WRKAUT  
WRKOBJOWN  
WRKOBJPGP

Aplicam-se as mesmas restrições aos ecrãs do utilizador que as descritas em “Aceder através de menus e ecrãs” na página 61.

**Utilizar o comando MKDIR no sistema de ficheiros QNTC:** Utilize o comando Criar Directório (MKDIR) para adicionar um directório do servidor ao directório /QNTC. É criado automaticamente um directório QNTC para todos os servidores funcionais do Windows no domínio e sub-rede local do iSeries NetServer. Esses servidores baseados no Windows exteriores à sub-rede local ou ao domínio do iSeries NetServer têm de ser adicionados utilizando o comando MKDIR ou a API mkdir(). Por exemplo:

```
MKDIR '/QNTC/NTSRV1'
```

adicionaria o servidor NTSRV1 à estrutura de directórios do sistema de ficheiros QNTC, de modo a permitir o acesso de ficheiros e directórios nesse servidor.

Também pode adicionar um novo servidor à estrutura de directórios utilizando o endereço de TCP/IP. Por exemplo:

```
MKDIR '/QNTC/9.130.67.24'
```

adicionaria o servidor à estrutura de directórios do sistema de ficheiros QNTC.

**Nota:** A configuração do iSeries NetServer for WINS permite a criação automática de directórios para servidores exteriores à sub-rede.

**Nota:** Se utilizar uma API mkdir() ou o comando de CL MKDIR para adicionar directórios à estrutura de directórios, estes não permanecerão visíveis entre IPLs. O comando MKDIR ou a API mkdir() têm de ser emitidos novamente após qualquer IPL do sistema.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNTC:** As APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros QNTC, com as seguintes excepções:

- As funções chmod(), fchmod(), utime() e umask() não terão qualquer efeito sobre os objectos do QNTC, mas se tentar utilizá-las, não causará um erro.
- O sistema de ficheiros QNTC não suporta as seguintes funções:
  - chown()
  - fchown()
  - givedescriptor()
  - link()
  - QjoEndJournal()

QjoRetrieveJournalInformation()  
QJORJIDI()  
QJOSJRNE()  
QjoStartJournal()  
Qp0lGetPathFromFileID()  
readlink()  
symlink()  
takedescriptor()

**Activar Kerberos para utilização com o sistema de ficheiros QNTC:** O QNTC permite o acesso do iSeries aos servidores baseados no Windows que suportam o protocolo de autenticação Kerberos V5. Em vez de utilizar uma palavra-passe do tipo gestor da Rede Local para obter a autenticação de cada servidor, um iSeries devidamente configurado pode agora aceder aos servidores CIFS suportados, através de uma única operação de início de sessão.

Antes de activar o Kerberos para utilização com o sistema de ficheiros QNTC, primeiro tem de configurar o seguinte:

- Network authentication service
- Enterprise Identity Mapping (EIM)

Após a configuração dos itens referidos acima, o utilizador pode usar o Kerberos com o sistema de ficheiros QNTC. Os passos seguintes são necessários para que um utilizador tire partido do suporte Kerberos QNTC.

- O perfil do utilizador de iSeries tem de ter o parâmetro de palavra-passe local, LCLPWDMGT, definido para \*NO. Especificando \*NO, o utilizador não terá uma palavra-passe de acesso ao sistema e não poderá iniciar sessão numa sessão 5250. O único acesso possível ao sistema é através das aplicações activadas de Kerberos, como o iSeries Navigator.

Se o utilizador especificar \*YES, a palavra-passe será gerida pelo sistema e o utilizador será autenticado sem Kerberos.

- Tem de ter uma senha do Kerberos e uma ligação ao iSeries Navigator.
- A senha Kerberos para o iSeries que está a utilizar tem de ser transmissível. Para tornar uma senha transmissível, siga os passos seguintes:
  - Aceda à ferramenta 'Utilizadores e computadores do Active Directory' em KDC para o domínio Kerberos
  - Seleccione os utilizadores
  - Seleccione o nome correspondente ao nome principal do serviço
  - Seleccione Propriedades
  - Seleccione o separador Conta
  - Nas opções de Conta seleccione 'A conta é fidedigna para delegação'

## **Sistema de ficheiros do Servidor de Ficheiros do OS/400 (QFileSvr.400)**

O sistema de ficheiros do servidor de ficheiros do OS/400 fornece um acesso transparente a outros sistemas de ficheiros que residam em servidores iSeries remotos. É acedido através de uma estrutura de directórios hierárquica.

O sistema de ficheiros QFileSvr.400 pode ser considerado como um cliente que age em nome dos utilizadores para executar pedidos de ficheiros. O QFileSvr.400 interage com o servidor de ficheiros do OS/400 no sistema destino para executar a operação de ficheiro efectiva.

Para obter mais informações sobre o QFileSvr.400, consulte Utilizar o QFileSvr.400 através da interface do sistema de ficheiros integrado.

## Utilizar o QFileSvr.400 através da interface do sistema de ficheiros integrado

O sistema de ficheiros QFileSvr.400 pode ser acedido através da interface do sistema de ficheiros integrado, utilizando o servidor de ficheiros do OS/400 ou comandos, ecrãs do utilizador e APIs do sistema de ficheiros integrado. Quando utilizar as interfaces do sistema de ficheiros integrado, deve ter em atenção as considerações e limitações que se seguem.

**Nota:** As características do sistema de ficheiros QFileSvr.400 são determinadas pelas características do sistema de ficheiros a ser acedido no servidor destino.

Os tópicos que se seguem fornecem mais informações sobre o sistema de ficheiros QFileSvr.400.

- “Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros do servidor de ficheiros do OS/400”
- “Nomes de caminhos no sistema de ficheiros do servidor de ficheiros do OS/400”
- “Comunicações no sistema de ficheiros do servidor de ficheiros do OS/400” na página 55
- “Segurança e autoridade de objectos no sistema de ficheiros do servidor de ficheiros do OS/400” na página 56
- “Ligações no sistema de ficheiros do servidor de ficheiros do OS/400” na página 56
- “Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros do servidor de ficheiros do OS/400” na página 56
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros do servidor de ficheiros do OS/400” na página 57

### Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros do servidor de ficheiros do OS/400:

Para um directório de primeiro nível, que na realidade representa o directório “raiz” (/) do sistema destino, o sistema de ficheiros QFileSvr.400 mantém o mesmo formato de maiúsculas e minúsculas no qual os nomes de objectos são introduzidos. No entanto, não é feita qualquer distinção entre maiúsculas e minúsculas quando o QFileSvr.400 procura nomes.

Para todos os outros directórios, a sensibilidade a maiúsculas e minúsculas depende do sistema de ficheiros específico a ser acedido. O QFileSvr.400 mantém o mesmo formato de maiúsculas e minúsculas no qual os nomes de objectos são introduzidos, quando os pedidos de ficheiros são enviados para o servidor de ficheiros do OS/400.

### Nomes de caminhos no sistema de ficheiros do servidor de ficheiros do OS/400:

- Os nomes de caminho têm o seguinte formato:

`/QFileSvr.400/NomeLocalizaçãoRemota/Directório/Directório . . . /Objecto`

O directório de primeiro nível (isto é, NomeLocalizaçãoRemota no exemplo anterior) representa ambas as seguintes opções:

- O nome do servidor destino que será utilizado para estabelecer uma ligação de comunicações. O nome do servidor destino pode ser um dos seguintes:
  - O nome de um sistema central de TCP/IP (por exemplo, `beowulf.newyork.corp.com`)
  - Um nome LU 6.2 de SNA (por exemplo, `appn.newyork`).
- O directório “raiz” (/) do servidor destino

Assim, quando é criado um directório de primeiro nível utilizando uma interface do sistema de ficheiros integrado, são ignorados quaisquer atributos especificados.

**Nota:** Os directórios de primeiro nível não são mantidos entre diversos IPLs. Ou seja, os directórios de primeiro nível têm de ser novamente criados após cada IPL.

- Cada componente do nome de caminho pode ter até 255 caracteres de comprimento. O nome completo do caminho pode ter até 16 megabytes de comprimento.

**Nota:** O sistema de ficheiros no qual o objecto reside pode restringir o comprimento do componente e o comprimento do nome de caminho para um valor inferior ao máximo permitido pelo QFileSvr.400.

- Não existem limites à profundidade da hierarquia de directórios, a não ser os limites de programa e de sistema, bem como quaisquer limites impostos pelo sistema de ficheiros a ser acedido.
- Os caracteres nos nomes são convertidos para o formato UCS2 Nível 1 quando os nomes são armazenados (consulte “Continuidade de nomes” na página 17).

#### **Comunicações no sistema de ficheiros do servidor de ficheiros do OS/400:**

- As ligações de TCP ao servidor de ficheiros num servidor destino só podem ser estabelecidas se o subsistema QSERVER no servidor destino estiver activo.
- As ligações LU 6.2 de SNA só são possíveis se existir uma sessão controlada localmente que não esteja em utilização (por exemplo, uma sessão especificamente estabelecida para ser utilizada pela ligação LU 6.2). Quando estabelece ligações LU 6.2, o sistema de ficheiros QFileSvr.400 utiliza um modo BLANK. No sistema destino é submetido ao subsistema QSERVER um trabalho designado QPWFSERV. O perfil do utilizador deste trabalho é definido pela entrada de comunicações para o modo BLANK. Para obter

mais informações sobre comunicações LU 6.2, consulte a publicação APPC Programming .

- Os pedidos do servidor de ficheiros que utilizam TCP como o protocolo de comunicações são executados no contexto do trabalho que está a emitir o pedido. Os pedidos do servidor de ficheiros que utilizam SNA como protocolo de comunicações são executados pelo trabalho Q400FILSVR do sistema OS/400.
- Se ainda não tiver sido estabelecida ligação ao servidor destino, o sistema de ficheiros QFileSvr.400 assume que o directório de primeiro nível representa um nome de sistema central de TCP/IP. Para estabelecer uma ligação ao servidor destino, o sistema de ficheiros QFileSvr.400 executa os seguintes passos:
  1. Atribui o nome da localização remota a um endereço de IP.
  2. Liga ao programa de definição de correspondências do servidor do sistema central na porta 449 já conhecida utilizando o endereço de IP resolvido. Em seguida, envia ao programa de definição de correspondências do servidor uma consulta do nome do serviço “como-ficheiro.” Uma das seguintes situações ocorre como resultado da consulta:
    - Se “como-ficheiro” estiver na tabela de serviços do servidor destino, o programa de definição de correspondências do servidor devolve a porta onde o daemon do servidor de ficheiros do OS/400 está a aguardar sinal.
    - Se o programa de definição de correspondências do servidor não estiver activo no servidor destino, será utilizado o número de porta assumido de “como-ficheiro” (8473).

Em seguida, o sistema de ficheiros QFileSvr.400 tenta estabelecer uma ligação TCP ao daemon do servidor de ficheiros do OS/400 no servidor destino. Quando a ligação for estabelecida, o QFileSvr.400 troca pedidos e respostas com o servidor de ficheiros. No subsistema QSERVER, os pedidos iniciados previamente QPWFSERVSO têm o controlo da ligação. Cada trabalho iniciado previamente é executado sob o respectivo perfil de utilizador.

3. Se o nome da localização remota não for resolvido como um endereço de IP, o directório de primeiro nível é considerado como um nome de LU 6.2 de SNA. Assim, é feita uma tentativa para estabelecer uma ligação APPC com o servidor de ficheiros do OS/400.
- O sistema de ficheiros QFileSvr.400 verifica periodicamente (de 2 em 2 horas) se existem ligações que não estejam a ser utilizadas (por exemplo, se não existem ficheiros abertos associados à ligação) e se essas ligações não tiveram actividade durante um período de 2 horas. Se for encontrada uma ligação deste tipo, a ligação é terminada.
  - O sistema de ficheiros QFileSvr.400 não pode detectar ciclos. O seguinte nome de caminho é um exemplo de um ciclo:

```
/QFileSvr.400/Remoto2/QFileSvr.400/Remoto1/QFileSvr.400/Remoto2/...
```

onde Remoto1 é o sistema local. Quando é especificado o nome de caminho que contém um ciclo, o sistema de ficheiros QFileSvr.400 devolve um erro após um breve período de tempo. O erro indica que o tempo de espera foi excedido.

O sistema de ficheiros QFileSvr.400 utilizará uma sessão livre existente quando comunicar através de SNA. É necessário iniciar o modo e estabelecer uma sessão para o QFileSvr.400 ligar com êxito ao sistema de comunicações remoto.

**Segurança e autoridade de objectos no sistema de ficheiros do servidor de ficheiros do OS/400:** Se ambos os sistemas tiverem o Kerberos configurado e o utilizador tiver sido autenticado para Kerberos, o Kerberos pode ser utilizado para autenticar um sistema de ficheiros que resida num servidor iSeries destino. Se a autenticação Kerberos falhar, o ID de utilizador e a palavra-passe podem ser utilizados para verificar o acesso.

**Nota:** Se a senha de concessão de senhas ou a senha do servidor expirar após o servidor destino ter verificado o seu acesso, a expiração só será efectiva quando a ligação ao servidor destino tiver terminado. Para obter mais informações sobre Kerberos, consulte o tópico Network authentication service do iSeries Information Center.

- Para ter acesso a um sistema de ficheiros residente num iSeries server destino, terá de ter um ID de utilizador e uma palavra-passe no servidor destino que correspondam ao ID de utilizador e à palavra-passe no servidor local, caso o Kerberos não seja utilizado para autenticação.

**Nota:** Se a sua palavra-passe no servidor local ou destino for alterada depois de o servidor destino ter verificado o seu acesso, a alteração não será reflectida até a ligação ao servidor destino ter terminado. No entanto, não haverá atraso se o perfil do utilizador no servidor local for eliminado e for criado outro perfil de utilizador com o mesmo ID de utilizador. Neste caso, o sistema de ficheiros QFileSvr.400 verifica se tem acesso ao servidor destino.

- A autoridade sobre objectos baseia-se no perfil de utilizador que reside no servidor destino. Ou seja, só lhe é permitido aceder a um objecto do sistema de ficheiros no servidor destino se o seu perfil de utilizador no servidor destino tiver a autoridade adequada sobre o objecto.

**Ligações no sistema de ficheiros do servidor de ficheiros do OS/400:** O sistema de ficheiros QFileSvr.400 suporta apenas uma ligação a um objecto. Não é possível criar nem armazenar ligações simbólicas no QFileSvr.400. No entanto, os ficheiros do QFileSvr.400 podem ser acedidos utilizando uma ligação simbólica a partir do sistema de ficheiros "raiz" (/), QOpenSys ou de sistemas de ficheiros definidos pelo utilizador.

Consulte "Ligação" na página 11 para obter uma descrição das ligações.

**Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros do servidor de ficheiros do OS/400:** Os comandos incluídos em "Aceder através de comandos de CL" na página 62 podem funcionar no sistema de ficheiros do QFileSvr.400, com as seguintes excepções:

ADDLNK  
APYJRNCHG  
CHGAUT  
CHGJRNOBJ  
CHGOWN  
DSPAUT  
ENDJRN  
RST  
SAV  
SNDJRNE  
STRJRN

WRKOBJOWN  
WRKOBJPGP

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder através de menus e ecrãs” na página 61.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros do servidor de ficheiros do OS/400:** As APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros QFileSvr.400, com as seguintes excepções:

chown()  
fchown()  
fclear()  
fclear64()  
givedescriptor()  
link()  
QjoEndJournal()  
QjoRetrieveJournalInformation()  
QJORJIDI()  
QJOSJRNE  
QjoStartJournal  
Qp0lGetPathFromFileID()  
symlink()  
takedescriptor()

## Sistema de ficheiros da rede (NFS)

O sistema de ficheiros NFS permite ao utilizador aceder a dados e objectos armazenados num servidor de NFS remoto. Um servidor de NFS pode exportar um sistema de ficheiros da rede que será instalado dinamicamente pelos clientes de NFS.

Além disso, qualquer sistema de ficheiros instalado localmente através do Sistema de Ficheiros da Rede terá as funções, características, limitações e dependências do directório ou sistema de ficheiros a partir do qual foi instalado no servidor remoto. As operações em sistemas de ficheiros instalados não são executadas localmente. Os pedidos passam através da ligação ao servidor e têm de respeitar os requisitos e as restrições do tipo de sistema de ficheiros do servidor.

Para obter mais informações sobre NFS, consulte Utilizar sistemas de ficheiros NFS através da interface do sistema de ficheiros integrado.

### Utilizar sistemas de ficheiros NFS através da interface do sistema de ficheiros integrado

O Sistema de Ficheiros da Rede (ou NFS) pode ser acedido através da interface do sistema de ficheiros integrado e tem as seguintes considerações e limitações.

Os tópicos que se seguem fornecem mais informações sobre o sistema de ficheiros NFS.

- “Características do sistema de ficheiros da rede” na página 58
- “Variações de servidores e clientes no sistema de ficheiros da rede” na página 58
- “Ligações no sistema de ficheiros da rede” na página 58
- “Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros de rede” na página 58
- “Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros de rede” na página 60

**Características do sistema de ficheiros da rede:** As características de qualquer sistema de ficheiros instalado através do NFS dependem do tipo de sistema de ficheiros que foi instalado a partir do servidor. É importante compreender que os pedidos executados no que parece ser um directório ou sistema de ficheiros local estão, de facto, a ser executados no servidor, através da ligação NFS.

Esta relação cliente/servidor pode ser confusa. Suponha, por exemplo, que instalou o sistema de ficheiros QDLS a partir do servidor, sobre um ramo do directório "raiz" (/) do seu cliente. Apesar de o sistema de ficheiros instalado parecer ser uma extensão do directório local, funciona e trabalha, de facto, como o sistema de ficheiros QDLS.

A compreensão desta relação para os sistemas de ficheiros instalados através do NFS é importante para o processamento de pedidos, quer localmente quer através da ligação ao servidor. O facto de um comando ser processado correctamente a nível local não significa que funcionará no directório instalado a partir do servidor. Cada directório instalado no cliente terá as propriedades e características do sistema de ficheiros do servidor.

**Variações de servidores e clientes no sistema de ficheiros da rede:** Existem três grandes possibilidades para as ligações cliente/servidor que podem afectar a forma como o Network File System funcionará e as características que terá:

1. O utilizador instala um sistema de ficheiros a partir de um servidor iSeries num cliente.
2. O utilizador instala um sistema de ficheiros a partir de um servidor UNIX num cliente.
3. O utilizador instala um sistema de ficheiros num cliente a partir de um servidor sem ser iSeries e sem ser UNIX.

No primeiro cenário, o sistema de ficheiros instalado no cliente terá um comportamento semelhante ao demonstrado no iSeries server. No entanto, ambas as características do Sistema de Ficheiros de Rede e do sistema de ficheiros que está a ser utilizado têm de ser tomadas em consideração. Por exemplo, se instalar o sistema de ficheiros QDLS no cliente a partir do servidor, ele terá as mesmas características e limitações que o sistema de ficheiros QDLS. Por exemplo, no sistema de ficheiros QDLS, os componentes do nome de caminho estão limitados a 8 caracteres, mais uma extensão de 3 caracteres. No entanto, o sistema de ficheiros instalado também terá características e limitações inerentes a NFS. Por exemplo, não pode utilizar o comando CHGAUD para alterar o valor de auditoria de um objecto NFS.

No segundo cenário, é importante perceber que qualquer sistema de ficheiros instalado a partir de um servidor UNIX terá um comportamento muito semelhante ao do sistema de ficheiros QOpenSys do servidor iSeries. Para obter mais informações sobre o sistema de ficheiros QOpenSys, consulte "Sistema de ficheiros de sistemas abertos (QOpenSys)" na página 30.

No terceiro cenário, terá de consultar a documentação sobre o sistema de ficheiros associado ao sistema operativo do servidor.

**Ligações no sistema de ficheiros da rede:** Em geral, são permitidas múltiplas ligações permanentes ao mesmo objecto no Sistema de Ficheiros da Rede. As ligações simbólicas são completamente suportadas. Pode ser utilizada uma ligação simbólica para ligar um Sistema de Ficheiros da Rede a um objecto de outro sistema de ficheiros. As capacidades de múltiplas ligações permanentes e simbólicas dependem por completo do sistema de ficheiros que estiver a ser instalado com o NFS.

Consulte "Ligação" na página 11 para obter uma descrição das ligações.

**Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros de rede:** Todos os comandos incluídos em "Aceder através de comandos de CL" na página 62 e os ecrãs descritos em "Aceder através de menus e ecrãs" na página 61 podem funcionar no Sistema de Ficheiros de Rede, com excepção dos seguintes:

- APYJRNCHG
- CHGJRNOBJ

- CHGAUD
- CHGATR
- CHGAUT
- CHGOWN
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN

Existem alguns comandos de CL específicos para o Sistema de Ficheiros da Rede e outros sistemas de ficheiros em geral. No entanto, a utilização destes comandos num processo que suporte múltiplos módulos pode não ser segura. A tabela seguinte descreve esses comandos. Para obter uma descrição completa dos comandos e ecrãs directamente relacionados com o Sistema de Ficheiros da Rede, consulte

OS/400 Network File System Support  .

*Tabela 6. Comandos de CL do Sistema de Ficheiros da Rede*

Comando	Descrição
ADDMFS	Adicionar Sistema de Ficheiros Instalado. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais.
CHGNFSEXP	Alterar Sistema de Ficheiros da Rede. Exporta, adiciona ou remove árvores de directórios para a tabela de exportações de sistemas de ficheiros exportados para clientes do Sistema de Ficheiros da Rede.
DSPMFSINF	Ver Informações sobre o Sistema de Ficheiros Instalado. Mostra informações sobre um sistema de ficheiros instalado.
ENDNFSSVR	Terminar Servidor do Sistema de Ficheiros da Rede. Termina um ou todos os daemons do Sistema de Ficheiros da Rede no servidor.
EXPORTFS	Exportar um Sistema de Ficheiros. Adiciona ou remove árvores de directórios para a tabela de exportações de sistemas de ficheiros exportados para clientes do Sistema de Ficheiros da Rede.
MOUNT	Instalar um Sistema de Ficheiros. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais. Este comando é uma alternativa ao comando ADDMFS.
RLSIFSLCK	Libertar Bloqueios do Sistema de Ficheiros Integrado. Liberta todos os bloqueios no intervalo de bytes do Sistema de Ficheiros da Rede suportados por um cliente ou de um objecto.
RMVMFS	Remover Sistema de Ficheiros Instalado. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local.
STRNFSSVR	Iniciar Servidor do Sistema de Ficheiros da Rede. Inicia um ou todos os daemons do Sistema de Ficheiros da Rede no servidor.
UNMOUNT	Desinstalar um Sistema de Ficheiros. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local. Este comando é alternativo ao comando RMVMFS.

**Nota:** É necessário instalar um Sistema de Ficheiros da Rede para que seja possível utilizar comandos no mesmo.

**Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros de rede:** Todas as APIs incluídas em “Executar operações utilizando APIs” na página 98 podem funcionar no sistema de ficheiros da rede, à excepção das seguintes:

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

Para obter uma descrição completa das funções de linguagem C que estão directamente relacionadas com o Sistema de Ficheiros da Rede, consulte OS/400 Network File System Support .

**Nota:** É necessário instalar um Sistema de Ficheiros da Rede para que seja possível utilizar APIs no mesmo.

---

## Aceder ao sistema de ficheiros integrado

Todas as interfaces de utilizador, tais como menus, comandos e ecrãs, que são utilizadas para trabalhar com bibliotecas, objectos, ficheiros de base de dados, pastas e documentos do sistema, continuam a funcionar tal como funcionavam antes da introdução do sistema de ficheiros integrado. No entanto, estas interfaces não podem ser utilizadas para trabalhar com os ficheiros de dados contínuos, directórios e outros objectos suportados pelo sistema de ficheiros integrado.

É fornecido um outro conjunto de interfaces de utilizador para o sistema de ficheiros integrado. Estas interfaces podem ser utilizadas em objectos de qualquer sistema de ficheiros acessível através dos directórios do sistema de ficheiros integrado.

Pode interagir com os directórios e objectos do sistema de ficheiros integrado a partir do servidor, através da utilização de menus e ecrãs ou de comandos da linguagem de controlo (CL). Além disso, pode utilizar interfaces de programas de aplicações (APIs) para tirar partido dos ficheiros de dados contínuos, directórios e outro suporte do sistema de ficheiros integrado.

Pode também interagir com o sistema de ficheiros integrado através do iSeries Navigator, uma interface gráfica de utilizador usada para gerir e administrar o servidor a partir do ambiente de trabalho do Windows.

Existem várias formas de interagir com o sistema de ficheiros integrado:

### **Aceder utilizando menus e ecrãs**

Pode executar operações em ficheiros e outros objectos no sistema de ficheiros integrado utilizando um conjunto de menus e ecrãs fornecidos pelo servidor.

### **Aceder utilizando comandos de CL**

Os comandos de CL podem funcionar em ficheiros e outros objectos de qualquer sistema de ficheiros que esteja acessível através do sistema de ficheiros integrado.

### **Aceder utilizando APIs**

As interfaces de programas de aplicações (APIs) que executam operações em directórios e ficheiros de dados contínuos do sistema de ficheiros integrado estão sob a forma de funções de linguagem C.

### **Aceder utilizando o iSeries Navigator**

O iSeries Navigator é a interface gráfica do utilizador que permite gerir e administrar os servidores a partir do ambiente de trabalho do Windows.

### **Aceder utilizando o iSeries NetServer**

Com o iSeries NetServer, os utilizadores do Windows podem aceder a caminhos de directórios e filas de output partilhados do OS/400.

### **Aceder utilizando programas de transferência de ficheiros**

Os programas de transferência de ficheiros transferem ficheiros de vários sistemas de ficheiros diferentes. Também transferem pastas e documentos encontrados no sistema de ficheiros QDLS.

### **Aceder através de um PC**

Um PC ligado ao iSeries permite a interacção com directórios e objectos armazenados no sistema de ficheiros integrado. Os directórios e os objectos aparecerão como se estivessem armazenados no PC.

## **Aceder através de menus e ecrãs**

Pode executar operações em ficheiros e outros objectos no sistema de ficheiros integrado utilizando um conjunto de menus e ecrãs fornecidos pelo servidor. Para visualizar menus do sistema de ficheiros integrado:

1. Inicie sessão no servidor.
2. Prima **Enter** para continuar.
3. Selecione a opção **Ficheiros, Bibliotecas e Pastas** a partir do menu principal do iSeries.
4. Selecione a opção **Sistema de Ficheiros Integrado** no menu Ficheiros, Bibliotecas e Pastas.

A partir daqui, pode trabalhar com comandos de Directório, de Objecto ou de Segurança no sistema de ficheiros integrado, conforme as necessidades. No entanto, se souber qual é o comando de CL que vai utilizar, pode escrevê-lo na linha de comandos existente na parte inferior do ecrã e premir **Enter**, ignorando o menu de opções.

Além disso, pode aceder ao sistema de ficheiros integrado a partir de qualquer menu do servidor, executando os passos seguintes:

1. Escreva **GO DATA** em qualquer linha de comandos para visualizar o menu Ficheiros, Bibliotecas e Pastas.
2. Selecione a opção Sistema de ficheiros integrado.

Para ver um menu de comandos do Sistema de Ficheiros da Rede, escreva **GO CMDNFS** em qualquer linha de comandos. Para ver um menu de comandos do sistema de ficheiros definido pelo utilizador, escreva **GO CMDUDFS** em qualquer linha de comandos.

Nos menus do sistema de ficheiros integrado pode solicitar a apresentação de ecrãs onde pode efectuar as seguintes operações:

- Criar, converter e remover um directório
- Visualizar e alterar o nome do directório actual
- Adicionar, visualizar, alterar e remover ligações de objectos
- Copiar, mover e mudar o nome de objectos
- Dar saída e entrada a objectos
- Guardar (cópia de segurança) e restaurar objectos
- Visualizar e alterar proprietários dos objectos e autoridades do utilizador
- Visualizar e alterar atributos de objectos
- Copiar dados entre ficheiros de dados contínuos e membros de ficheiros de bases de dados
- Criar, eliminar e visualizar o estado de sistemas de ficheiros definidos pelo utilizador
- Exportar sistemas de ficheiros a partir de um servidor
- Instalar e remover sistemas de ficheiros num cliente

Alguns sistemas de ficheiros não suportam todas estas operações. Para obter informações sobre as restrições de determinados sistemas de ficheiros, consulte *Trabalhar com sistemas de ficheiros*.

Consulte o tópico que se segue para obter mais informações sobre menus e ecrãs do sistema de ficheiros integrado:

- Regras de nomes de caminho para comandos de CL e ecrãs
- Aceder através de comandos de CL

## Aceder através de comandos de CL

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado (consulte “Aceder através de menus e ecrãs” na página 61) podem ser efectuadas introduzindo comandos de linguagem de controlo (CL). Estes comandos podem ser utilizados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através do interface do sistema de ficheiros integrado.

A Tabela 1 resume os comandos do sistema de ficheiros integrado. Para obter mais informações sobre comandos de CL especificamente relacionados com sistemas de ficheiros definidos pelo utilizador, o Sistema de Ficheiros da Rede e sistemas de ficheiros instalados em geral, consulte “Sistema de ficheiros definido pelo utilizador (UDFS)” na página 32 e “Sistema de ficheiros da rede (NFS)” na página 57. Sempre que um comando execute a mesma operação que um comando do OS/2 ou do DOS, é fornecido um nome de comando alternativo para comodidade dos utilizadores do OS/2 e do DOS.

*Tabela 7. Comandos do Sistema de Ficheiros Integrado*

Comando	Descrição	Nome alternativo
ADDLNK	Adicionar Ligação. Adiciona uma ligação entre um directório e um objecto.	
ADDMFS	Adicionar Sistema de Ficheiros Instalado. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais.	MOUNT
APYJRNCHG <sup>2</sup>	Aplicar Alterações Registadas em Diário. Utiliza entradas de diário para aplicar alterações ocorridas desde que um objecto registado em diário foi guardado ou para aplicar alterações até um determinado ponto.	
CHGATR	Alterar Atributo. Altera um atributo para um único objecto, para um grupo de objectos ou para uma árvore de directórios onde o directório, o respectivo conteúdo e o conteúdo de todos os subdirectórios nele incluídos que têm o atributo alterado.	
CHGAUD	Alterar Valor de Auditoria. Activa ou desactiva a auditoria de um objecto.	
CHGAUT	Alterar Autoridade. Concede uma autoridade específica para um objecto a um utilizador ou a um grupo de utilizadores.	
CHGCURDIR	Alterar Directório Actual. Altera o directório para ser utilizado como directório actual.	CD, CHDIR
CHGJRNOBJ <sup>2</sup>	Alterar Objectos Registados em Diário. Altera os atributos de registo em diário de um objecto ou lista de objectos sem ter de terminar e reiniciar o registo em diário do objecto.	
CHGNFSEXP	Alterar Exportação do Sistema de Ficheiros da Rede. Adiciona árvores de directórios ou remove-as da tabela de exportação que é exportada para os clientes de NFS.	EXPORTFS
CHGOWN	Alterar Proprietário. Transfere a propriedade de um objecto de um utilizador para outro.	
CHGPGP	Alterar Grupo Primário. Altera o grupo primário de um utilizador para outro.	

Tabela 7. Comandos do Sistema de Ficheiros Integrado (continuação)

Comando	Descrição	Nome alternativo
CHKIN	Dar Entrada. Dá entrada a um objecto ao qual previamente tinha sido dada saída.	
CHKOUT	Dar Saída. Dá saída a um objecto, o que impede que seja alterado por outros utilizadores.	
CPY	Copiar. Copia um único objecto ou um grupo de objectos.	COPY
CPYFRMSTMF	Copiar de Ficheiro de Dados Contínuos. Copia dados de um ficheiro de dados contínuos para um membro de ficheiro de base de dados.	
CPYTOSTMF	Copiar para Ficheiro de Dados Contínuos. Copia dados de um membro de ficheiro de base de dados para um ficheiro de dados contínuos.	
CRTDIR	Criar Directório. Adiciona um novo directório ao sistema.	MD, MKDIR
CRTUDFS	Criar UDFS. Cria um Sistema de Ficheiros Definido pelo Utilizador.	
CVTDIR	Converter Directório. Fornece informações acerca da conversão de directórios do sistema de ficheiros integrado do formato *TYPE1 para o formato *TYPE2.	
CVTRPCSRC	Converter Origem de RPC. Gera código C de um ficheiro de entrada escrito na linguagem de Chamada de Procedimento Remoto (RPC).	RPCGEN
DLTUDFS	Eliminar UDFS. Elimina um Ficheiro Definido pelo Utilizador.	
DSPAUT	Visualizar Autoridade. Mostra uma lista de utilizadores autorizados de um objecto e as respectivas autoridades para o mesmo.	
DSPCURDIR	Visualizar Directório Actual. Mostra o nome do directório actual.	
DSPLNK	Visualizar Ligações de Objecto. Mostra uma lista de objectos num directório e fornece opções de visualização das informações referentes aos objectos.	
DSPF	Ver Ficheiro de Dados Contínuos. Apresenta um ficheiro de dados contínuos ou um ficheiro de base de dados.	
DSPMFSINF	Visualizar Informações sobre o Sistema de Ficheiros Instalado. Apresenta informações acerca de um sistema de ficheiros instalado.	STATFS
DSPUDFS	Visualizar UDFS. Mostra o Sistema de Ficheiros Definido pelo Utilizador.	
EDTF	Editar Ficheiro de Dados Contínuos. Edita um ficheiro de dados contínuos ou um ficheiro de base de dados.	
ENDJRN <sup>2</sup>	Terminar Registo em Diário. Termina o registo em diário das alterações efectuadas a um objecto ou lista de objectos.	
ENDNFSSVR	Terminar Servidor do Sistema de Ficheiros da Rede. Termina um ou todos os daemons de NFS no servidor e no cliente.	
ENDRPCBIND	Terminar Daemon do Processador de Enlaces de RPC. Termina o daemon RPCBind de Chamada de Procedimento Remoto (RPC).	
MOV	Mover. Move um objecto para um directório diferente.	MOVE
PRDIRINF	Imprimir Informações do Directório. É utilizado para imprimir informações do directório relativas a objectos do sistema de ficheiros integrado recolhidas através do comando Obter Informações do Directório (RTVDIRINF).	

Tabela 7. Comandos do Sistema de Ficheiros Integrado (continuação)

Comando	Descrição	Nome alternativo
RLSIFSLCK	Libertar Bloqueios do Sistema de Ficheiros Integrado. Liberta todos os bloqueios no intervalo de bytes retidos por um cliente de NFS ou de um objecto.	
RMVDIR	Remover Directório. Remove um directório do sistema.	RD, RMDIR
RMVLNK	Remover Ligação. Remove a ligação a um objecto.	DEL, ERASE
RMVMFS	Remover Sistema de Ficheiros Instalado. Remove os sistemas de ficheiros exportados do servidor remoto dos directórios de cliente locais.	UNMOUNT
RNM	Mudar o nome. Altera o nome de um objecto num directório.	REN
RPCBIND	Iniciar Daemon de Processador de Enlaces de RPC. Inicia o Daemon RPCBind de Chamada de Procedimento Remoto (RPC).	
RST	Restaurar. Copia um objecto ou grupo de objectos de um dispositivo de cópia de segurança para o sistema.	
RTVCURDIR	Obter Directório Actual. Obtém o nome do directório actual e coloca-o numa variável especificada (utilizada em programas de CL)	
RTVDIRINF	Obter Informações do Directório. É utilizado para recolher atributos de objectos no sistema de ficheiros integrado.	
SAV	Guardar. Copia um objecto ou grupo de objectos do sistema para um dispositivo de cópia de segurança.	
SNDJRNE <sup>2</sup>	Enviar Entrada de Diário. Adiciona as entradas de diário do utilizador, associadas por opção a um objecto registado em diário, a um receptor de diário.	
STRJRN <sup>2</sup>	Iniciar Registos em Diário. Inicia o registo de alterações em diário (efectuadas num objecto ou numa lista de objectos) num diário específico.	
STRNFSSVR	Iniciar Servidor do Sistema de Ficheiros da Rede. Inicia um ou todos os daemons de NFS no servidor e no cliente.	
WRKAUT	Trabalhar com Autoridade. Mostra uma lista de utilizadores e as respectivas autoridades e fornece opções para adicionar um utilizador, alterar uma autoridade de utilizador ou remover um utilizador.	
WRKLNK	Trabalhar com Ligações de Objectos. Mostra uma lista de objectos num directório e fornece opções para executar acções nos objectos.	
WRKOBJOWN <sup>1</sup>	Trabalhar com Objectos por Proprietário. Mostra uma lista de objectos que são propriedade de um perfil de utilizador e fornece opções para executar acções nos objectos.	
WRKOBJPGP <sup>1</sup>	Trabalhar com Objectos por Grupo Primário. Mostra uma lista de objectos controlados por um grupo primário e fornece opções para executar acções nos objectos.	

**Nota:**

1. Os comandos WRKOBJOWN e WRKOBJPGP podem apresentar todos os tipos de objectos mas podem não ser completamente funcionais em todos os sistemas de ficheiros.
2. Consulte Journal management no iSeries Information Center para obter mais informações.

Consulte os tópicos que se seguem para obter mais informações acerca de restrições e comandos de CL do sistema de ficheiros integrado através da utilização destes comandos em determinados sistemas de ficheiros:

- Trabalhar com sistemas de ficheiros
- Regras de nomes de caminho para comandos de CL e ecrãs
- Trabalhar com output dos comandos RTVDIRINF e PRDIRINF
- O tópico CL do Centro de Informações do iSeries

## Regras de nomes de caminho para comandos de CL e ecrãs

Quando utilizar um comando ou ecrã do sistema de ficheiros integrado para funcionar com um objecto, identifique o objecto indicando o respectivo nome de caminho. A seguir encontra-se um resumo das regras, a não esquecer, quando especificar nomes de caminho. Nestas regras, o termo **objecto** refere-se a qualquer directório, ficheiro, ligação ou outro objecto.

- Os nomes dos objectos devem ser exclusivos dentro de cada directório.
- O nome de caminho que é transferido para um comando de CL do sistema de ficheiros integrado tem de ser representado no CCSID presentemente em vigor para o trabalho. Se o CCSID do trabalho for 65535, o nome de caminho tem de ser representado no CCSID assumido do trabalho. Uma vez que as cadeias de texto são habitualmente codificadas em CCSID 37, é necessário converter os nomes de caminho gravados no código para o CCSID do trabalho, antes de transferir o caminho para o comando.
- Os nomes de caminho têm de estar entre plicas (') quando introduzidos numa linha de comandos. Estas plicas são facultativas quando os nomes de caminho são introduzidos nos ecrãs. No entanto, se o nome de caminho incluir cadeias entre plicas, as plicas ' ' também têm de ser incluídas.
- Os nomes de caminho são introduzidos da esquerda para a direita, começando pelo directório com o nível mais elevado e terminando no nome do objecto a ser executado pelo comando. O nome de cada componente no caminho está separado por uma barra (/) ou por uma barra invertida (\); por exemplo:

```
'Dir1/Dir2/Dir3/FichUti'
```

ou

```
'Dir1\Dir2\Dir3\FichUti'
```

- Os caracteres / e \ e os nulos não podem ser utilizados nos componentes individuais do nome de caminho (uma vez que / e \ são utilizadas como separadores). As minúsculas não são alteradas para maiúsculas pelos comandos. O nome pode ou não ser alterado para maiúsculas, se o sistema de ficheiros que contém o objecto é sensível a maiúsculas e minúsculas e se o objecto está a ser criado ou procurado.
- O comprimento do nome de um objecto está limitado pelo sistema de ficheiros onde o objecto se encontra e pelo comprimento máximo de uma cadeia de comandos. Os comandos aceitam nomes de objecto até 255 caracteres de comprimento e nomes de caminho até 5000 caracteres.

Consulte Trabalhar com sistemas de ficheiros para ver os limites de nomes de caminho em cada sistema de ficheiros.

- Uma / ou \ no início de um nome de caminho significa que o caminho começa no directório mais elevado, o directório "raiz" (/); por exemplo:

```
'/Dir1/Dir2/Dir3/FichUti'
```

- Se o nome de caminho não começar por / ou \, assume-se que o caminho começa no directório actual do utilizador que introduz o comando; por exemplo:

```
'MeuDir/MeuFich'
```

em que MeuDir é um subdirectório do directório actual do utilizador.

- Um til (~) seguido de uma barra (ou barra invertida) no início de um nome de caminho significa que o caminho começa no directório inicial do utilizador que introduz o comando; por exemplo:

```
'~/DirUt1/ObjUt1'
```

- Um til (~) seguido de um nome do utilizador e, em seguida, de uma barra (ou barra invertida) no início de um nome de caminho significa que o caminho começa no directório inicial do utilizador identificado pelo nome do utilizador; por exemplo:

'~nome-utilizador/DirUt1/ObjUt1'

- Em alguns comandos, pode ser utilizado um asterisco (\*) ou um ponto de interrogação (?) no último componente do nome de caminho para procurar padrões de nomes. O \* indica ao sistema que procure nomes que contenham qualquer número de caracteres na posição do carácter \*. O ? indica ao sistema que procure nomes que tenham um único carácter na posição do ?. O exemplo que se segue procura todos os objectos cujos nomes comecem por *d* e terminem em *txt*:

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

O exemplo que se segue procura os objectos cujos nomes começam por *d*, seguidos de um único carácter, e terminem em *txt*:

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

- Para evitar uma possível confusão com valores especiais do servidor iSeries, os nomes de caminho não podem ser iniciados por um asterisco (\*). Para executar uma correspondência de padrões no início de um nome de caminho utilize dois asteriscos (\*\*); por exemplo:

'\*\*.ficheiro'

**Nota:** Esta regra só se aplica a nomes de caminho relativos em que não existem outros caracteres antes do asterisco (\*).

- Ao trabalhar com objectos no sistema de ficheiros QSYS.LIB, os nomes de componentes têm de estar no formato *nome.tipo-objecto*; por exemplo:

'/QSYS.LIB/PAG.LIB/IMP.FILE'

Consulte “Sistema de ficheiros de biblioteca (QSYS.LIB)” na página 37 para obter mais detalhes.

- Ao trabalhar com objectos no sistema de ficheiros QSYS.LIB de ASP independente, os nomes de componentes têm de estar no formato *nome.tipo-objecto*; por exemplo:

'/nome\_asp/QSYS.LIB/PAYDAVE.LIB/PAY.FILE'

Consulte “QSYS.LIB de ASP independente” na página 40 para obter mais detalhes.

- O nome de caminho tem de estar entre conjuntos adicionais de plicas (') ou aspas ("), caso seja utilizado algum dos seguintes caracteres num nome de componente:

- Asterisco (\*)
- Ponto de interrogação (?)
- Plica (')
- Aspas (")
- Til (~), se for utilizado como primeiro carácter no nome do primeiro componente do nome de caminho (se for utilizado noutra posição, o til é interpretado como qualquer outro carácter)

Por exemplo:

'"/Dir1/Dir/A\*Silva''

ou

'''/Dir1/Dir/A\*Silva'''

Esta prática não é recomendada uma vez que o significado do carácter numa cadeia de comandos pode ser mal interpretado e é provável que a cadeia de comandos seja introduzida incorrectamente.

- Não utilize dois pontos (:) em nomes de caminho. Possui um significado especial no sistema.
- O suporte de processamento para comandos e ecrãs de utilizador associados não reconhece elementos de código inferiores ao hexadecimal 40 como sendo caracteres que possam ser utilizados em cadeias de comandos ou em ecrãs. Se estes elementos de código forem utilizados, têm de ser introduzidos como uma representação hexadecimal, tal como:

crtmdir dir(X'02')

Deste modo, não é recomendada a utilização de elementos de código inferiores ao hexadecimal 40 em nomes de caminho. Esta restrição só se aplica a comandos e ecrãs associados, não a APIs (consulte “Aceder utilizando APIs” na página 80). Além disso, não é permitido um valor hexadecimal 0 em nomes de caminho.

Para ver as restrições de utilização de um comando em particular, consulte a ajuda de comandos ou o tópico Control language (CL) no Centro de Informações do iSeries.

## Trabalhar com output dos comandos RTVDIRINF e PRTDIRINF

O comando RTVDIRINF é utilizado para recolher atributos de objectos no sistema de ficheiros integrado. As informações recolhidas são armazenadas em ficheiros de base de dados (tabelas) que são designados através do prefixo do ficheiro das informações especificado no parâmetro INFFILEPFX. As tabelas são criadas na biblioteca especificada pelo parâmetro INFLIB. Para obter mais informações sobre a denominação dos ficheiros da base de dados ou tabelas, consulte RTVDIRINF.

São criadas três tabelas como resultado do comando RTVDIRINF. Uma tabela armazena atributos de objectos, outra destina-se aos directórios e a última tabela é utilizada para determinar os ficheiros que foram utilizados para armazenar atributos de objectos.

O tópico “Aceder a dados de RTVDIRINF” na página 79 contém informações acerca de como pode aceder aos dados nestas três tabelas. Para obter mais informações sobre como utilizar os dados, consulte “Utilização dos dados de RTVDIRINF” na página 80.

A Tabela 8 descreve os campos facultados para a tabela que armazena os atributos dos objectos. Se especificar \*GEN no parâmetro INFFILEPFX, os ficheiros da base de dados serão criados com um prefixo exclusivo gerado por este comando. O prefixo começará por QAEZD seguido de quatro dígitos. Os ficheiros criados para armazenar as informações recolhidas serão denominados através deste prefixo seguido da letra 'D' (para o ficheiro que contém as informações do directório) ou da letra 'O' (para o ficheiro que contém as informações sobre os objectos nos directórios). Por exemplo, quando o comando for executado pela primeira vez e \*GEN tiver sido especificado, os ficheiros QAEZD0001D e QAEZD0001O serão criados na biblioteca especificada pelo parâmetro INFLIB. Os utilizadores podem especificar um prefixo de ficheiro até nove caracteres para ser utilizado na designação desta base de dados.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZDIRIDX	INTEGER	O índice remissivo do directório ascendente.
QEZOBJNAM <sup>1</sup>	VARGRAPHIC (1024)	O nome do objecto. <sup>2</sup>
QEZOBJLEN	INTEGER	O número de bytes existentes no campo do nome do objecto.
QEZNMCCSID	INTEGER	O CCSID em que é representado o nome do objecto.
QEZREGION	GRAPHIC (2)	Um ID com dois caracteres que representa o país do nome do objecto. (Afecta as acções cuja tendência é serem definidas pela localização da acção, tais como, a sequência de ordenação).
QEZLANGID	GRAPHIC (3)	Um ID com três caracteres que representa a linguagem em que se encontra o nome do objecto.
QEZMODE	INTEGER	O tipo e o modo de acesso ao ficheiro. Para obter mais informações acerca do modo, consulte a API Abrir ficheiro aberto().
QEZOBJTYPE <sup>1</sup>	GRAPHIC (10)	O tipo de objecto.
QEZCCSID	INTEGER	O CCSID dos dados e atributos expandidos do objecto.
QEZALCSIZE <sup>1</sup>	BIGINT	O número de bytes atribuídos a este objecto.
QEZDTASIZE	BIGINT	O tamanho em bytes dos dados deste objecto. Este tamanho não inclui os cabeçalhos do objecto ou o tamanho dos atributos expandidos associados ao objecto.
QEZEAS	BIGINT	Número de atributos expandidos associados a este objecto.
QEZCEAS	BIGINT	Número de atributos expandidos críticos associados a este objecto.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZEEXTATRS	BIGINT	Número total de bytes de todos os dados de atributos expandidos.
QEZCRTTIM	TIMESTAMP	A data e a hora de criação do objecto.
QEZACCTIM	TIMESTAMP	A data e a hora do último acesso aos dados do objecto.
QEZCHGTIMA <sup>1</sup>	TIMESTAMP	A data e a hora da última modificação dos atributos do objecto.
QEZCHGTIMD	TIMESTAMP	A data e a hora da última modificação dos dados do objecto.
QEZSTGFREE <sup>1</sup>	SMALLINT	Se os dados do objecto foram movidos para offline, libertando memória online. Os valores válidos são:  0 - Os dados do objecto não estão offline.  1 - Os dados do objecto estão offline.
QEZCHKOUT <sup>1</sup>	SMALLINT	Um indicador sempre que for dada saída a um objecto. Os valores válidos são:  0 - Não foi dada saída ao objecto.  1 - Foi dada saída ao objecto.
QEZCHKOWN	GRAPHIC (10)	O utilizador que deu saída ao objecto. Se não foi dada saída ao objecto, este campo estará em branco.
QEZCHKTIM	TIMESTAMP	A data e a hora de saída do objecto. Se não foi dada saída ao objecto, este campo não terá quaisquer dados.
QEZLOCAL	SMALLINT	Se um objecto é armazenado localmente ou num sistema remoto. Se um objecto é local ou remoto varia de acordo com as respectivas regras do sistema de ficheiros. Os objectos do sistema de ficheiros que não contenham um indicador remoto ou local são considerados remotos. Os valores válidos são:  1 - Os dados do objecto estão armazenados localmente.  2 - Os dados do objecto estão num sistema remoto.
QEZOWN <sup>1</sup>	GRAPHIC (10)	O nome do perfil de utilizador que é o proprietário do objecto ou o seguinte valor especial:  *NOUSRPRF - Este valor especial é utilizado pelo Sistema de Ficheiros da Rede para indicar que não existe um perfil de utilizador no servidor iSeries local com um ID de utilizador (UID) que corresponda ao UID do objecto remoto.
QEZUID	INTEGER	Todos os utilizadores do sistema têm de ter um número de identificação de utilizador numérico exclusivo (UID).
QEZOWNPGP	GRAPHIC (10)	O nome do perfil de utilizador que é o grupo primário do objecto ou os seguintes valores especiais:  *NONE - O objecto não tem um grupo primário.  *NOUSRPRF - Este valor especial é utilizado pelo Sistema de Ficheiros da Rede para indicar que não existe um perfil de utilizador no servidor local com um ID de grupo (GID) correspondente ao GID do objecto remoto.
QEZGID	INTEGER	A identificação dos perfis de grupo é efectuada através de um número de identificação de grupo (GID) numérico exclusivo.
QEZAUTLST	GRAPHIC (10)	O nome da lista de autorizações utilizada para proteger o objecto designado. O valor *NONE indica que não está a ser utilizada uma lista de autorizações para determinar a autoridade do objecto.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZASP	SMALLINT	O conjunto de memória auxiliar no qual está armazenado o objecto.
QEZJRNSTS <sup>1</sup>	SMALLINT	O estado actual do registo em diário do objecto. Este campo terá um dos valores que se seguem:  0 (NOT_JOURNALED) - Actualmente não está a ser efectuado o registo em diário do objecto.  1 (JOURNALED) - Actualmente está a ser efectuado o registo em diário do objecto.
QEZJSUBTRE	SMALLINT	Se o sinalizador for devolvido, este objecto é um directório com uma semântica de subárvore de registo em diário do sistema de ficheiros integrado.  0 - O registo em diário do objecto não é efectuado com uma semântica de subárvore.  1 - O registo em diário do objecto é efectuado com uma semântica de subárvore. Os objectos novos criados nesta subárvore do directório herdam os atributos e opções do registo em diário deste directório.
QEZJOPTENT	SMALLINT	Se o registo em diário estiver activo é efectuado o registo das entradas consideradas opcionais. A lista de entradas de diário opcionais varia consoante o tipo de objecto.  0 - O registo em diário do objecto não é efectuado com as entradas opcionais.  1 - O registo em diário do objecto é efectuado com as entradas opcionais.
QEZJAFTERI	SMALLINT	Se o registo em diário estiver activo, a imagem do objecto após uma alteração é registada.  0 - O registo em diário do objecto não é efectuado com imagens posteriores.  1 - O registo em diário do diário é efectuado com imagens posteriores.
QEZJBEFORI	SMALLINT	Se o registo em diário estiver activo, a imagem do objecto é registada no diário antes de uma alteração.  0 - O registo em diário do objecto não é efectuado com imagens anteriores.  1 - O registo em diário do objecto é efectuado com imagens anteriores.
QEZJRNID	GRAPHIC (10)	Este campo associa o objecto a ser registado em diário a um identificador que pode ser utilizado em vários comandos relacionados com o registo em diário e APIs.
QEZJRNNAM	GRAPHIC (10)	Se o valor do estado do registo em diário for JOURNALED, este campo contém o nome do diário utilizado actualmente. Se o valor do estado do registo em diário for NOT_JOURNALED, este campo contém o nome do último diário utilizado para este objecto. Todos os bytes deste campo serão definidos para zero binário se nunca foi efectuado o registo em diário deste objecto.
QEZJRNLIB	GRAPHIC (10)	Se o valor do estado do registo em diário for JOURNALED, este campo contém o nome da biblioteca com o diário utilizado actualmente. Se o valor do estado do registo em diário for NOT_JOURNALED, este campo contém o nome da biblioteca com o último diário utilizado. Todos os bytes deste campo serão definidos para zero binário se nunca foi efectuado o registo em diário deste objecto.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZJRNSTR	TIMESTAMP	Os segundos decorridos desde o início da fase correspondente à última data e hora de registo em diário do objecto. Este campo será definido para zero binário se nunca tiver sido efectuado o registo em diário deste objecto.
QEZAUDT	GRAPHIC (10)	O valor de auditoria associado ao objecto. Os valores válidos são:  *NONE - Não é efectuada uma auditoria a este objecto se o mesmo for lido ou alterado independentemente do utilizador que está a aceder ao objecto.  *USRPRF - Só efectuar uma auditoria a este objecto se estiver a ser efectuada uma auditoria ao utilizador actual. É efectuado um teste ao utilizador actual para determinar se deve ser efectuada uma auditoria a este objecto. É possível especificar através do perfil do utilizador se a auditoria é efectuada apenas ao acesso para alterações ou aos acessos para leitura e alterações a este objecto.  *CHANGE - Efectuar uma auditoria a todos os acessos para alterações a este objecto por todos os utilizadores do sistema.  *ALL - Efectuar uma auditoria a todos os acessos a este objecto por todos os utilizadores do sistema. Todos os acessos são definidos como uma operação de leitura ou alteração.
QEZBLKSIZ	INTEGER	O tamanho de bloco de um objecto.
QEZNLNK	INTEGER	O número de ligações permanentes ao objecto.
QEZFILEID <sup>1</sup>	GRAPHIC (16)	O ID de ficheiro do objecto. Um identificador associado ao objecto. É possível utilizar um ID de ficheiro com o Qp0lGetPathFromFileID() para obter um nome de caminho do objecto.
QEZFILEIDS	INTEGER	O ID de ficheiro de 4 bytes do ficheiro. Este número identifica unicamente o objecto num sistema de ficheiros. Não é possível através deste número identificar o objecto em todo o sistema.
QEZGENID	BIGINT	O ID de geração associado ao ID de ficheiro.
QEZFSID	BIGINT	O ID do sistema de ficheiros a que pertence o objecto. Este número identifica unicamente o sistema de ficheiros a que pertence o objecto.
QEZRDEV	BIGINT	Se o objecto representa um ficheiro especial de um dispositivo, representa o dispositivo real.
QEZDOM	GRAPHIC (10)	O domínio do objecto. Os valores válidos são:  *SYSTEM - O objecto existe no domínio do sistema.  *USER - O objecto existe no domínio do utilizador.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZCRTAUD	GRAPHIC (10)	<p>O valor de auditoria associado a um objecto criado neste directório. Os valores válidos são:</p> <p>*NONE - Não é efectuada uma auditoria a este objecto se o mesmo for lido ou alterado independentemente do utilizador que está a aceder ao objecto.</p> <p>*USRPRF - Só efectuar uma auditoria a este objecto se estiver a ser efectuada uma auditoria ao utilizador actual. É efectuado um teste ao utilizador actual para determinar se deve ser efectuada uma auditoria a este objecto. É possível especificar através do perfil do utilizador se a auditoria é efectuada apenas aos acessos para alterações ou aos acessos para leitura e alterações a este objecto.</p> <p>*CHANGE - Efectuar uma auditoria a todos os acessos para alterações a este objecto por todos os utilizadores do sistema.</p> <p>*ALL - Efectuar uma auditoria a todos os acessos a este objecto por todos os utilizadores do sistema. Todos os acessos são definidos como uma operação de leitura ou alteração.</p>
QEZSCN	GRAPHIC (1)	<p>Se o objecto será pesquisado quando os programas de saída são registados através de um dos pontos de saída relacionados com a leitura do sistema de ficheiros integrado. Para obter mais informações, consulte “Suporte de pesquisa” na página 18.</p> <p>Os valores válidos são:</p> <p>‘x40’ (SCANNING_NO) - O objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a leitura. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p> <p>‘x80’ (SCANNING_YES) - O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.</p> <p>‘x20’ (SCANNING_CHGONLY) - O objecto só será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado desde a última pesquisa. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado. Este atributo só entrará em vigor se o valor do sistema Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) tiver *USEOCOATR especificado. De outro modo, será considerado como se o atributo fosse SCANNING_YES. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p>

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZINHSCN	GRAPHIC (1)	<p>Se os objectos criados num directório serão pesquisados quando os programas de saída são registados através de um dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado. Para obter mais informações, consulte “Suporte de pesquisa” na página 18.</p> <p>Os valores válidos são:</p> <p>x’40’ - Depois de criado um objecto no directório, o objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p> <p>x’80’ - Após a criação de um objecto no directório, o objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o mesmo tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.</p> <p>x’20’ - Após a criação de um objecto no directório, o objecto só será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se tiver sido modificado desde a última pesquisa. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado. Este atributo só entrará em vigor se o valor do sistema Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) tiver *USEOCOATR especificado. De outro modo, será considerado como se o atributo fosse SCANNING_YES. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p>

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZSTATUS	GRAPHIC (1)	<p>O estado de pesquisa associado a este objecto. Este campo terá um dos valores que se seguem:</p> <p>x'00' (SCAN_REQUIRED) - É necessário efectuar uma pesquisa do objecto porque ainda não foi pesquisado pelos programas de saída relacionados com a pesquisa ou porque os dados dos objectos ou CCSID foram modificados desde a última pesquisa. Seguem-se exemplos de modificações de dados de objectos ou CCSID: escrever num objecto, directamente ou através da definição da memória; truncar o objecto; limpar o objecto; e alterar o atributo CCSID dos objectos.</p> <p>x'01' (SCAN_SUCCESS) - O objecto foi pesquisado através de um programa de saída relacionado com a pesquisa e no último pedido de pesquisa esta foi efectuada com êxito.</p> <p>x'02' (SCAN_FAILURE) - O objecto foi pesquisado através de um programa de saída relacionado com a pesquisa e no último pedido de pesquisa não foi possível efectuar a pesquisa e a operação não foi concluída. Se ocorreu uma falha num objecto, este só será pesquisado novamente se a assinatura de pesquisa do objecto for diferente da assinatura chave de pesquisa global ou da assinatura chave de pesquisa de grupo de ASPs independentes, conforme for adequada. Assim, não será possível executar os pedidos seguintes para trabalhar com o objecto e será apresentada uma indicação de falha na pesquisa. Exemplos de pedidos com falha são: abrir o objecto, alterar o CCSID do objecto, copiar o objecto.</p> <p>x'05' (SCAN_PENDING_CVN) - O objecto não está num directório *TYPE2 e, por conseguinte, só será pesquisado após a conversão do directório.</p> <p>x'06' (SCAN_NOT_REQUIRED) - Não é necessário efectuar uma pesquisa do objecto porque este está assinalado para não ser pesquisado.</p>
QEZSSIGDF	GRAPHIC (1)	<p>As assinaturas de pesquisa indicam o nível de suporte do software de pesquisa.</p> <p>Quando um objecto está num grupo de ASPs independentes, a assinatura de pesquisa do objecto é comparada à assinatura de pesquisa do grupo de ASPs independentes associado. Quando um objecto não está num grupo de ASPs independentes, a assinatura de pesquisa do objecto é comparada com o valor da assinatura de pesquisa global. Este campo terá um dos valores que se seguem:</p> <p>x'00' - As assinaturas comparadas não são diferentes.</p> <p>x'01' - As assinaturas comparadas são diferentes.</p>
QEZSBINARY	GRAPHIC (1)	<p>Indica se o objecto foi pesquisado no modo binário, quando foi pesquisado anteriormente. Este campo terá um dos valores que se seguem:</p> <p>x'00' - O objecto não foi pesquisado no modo binário.</p> <p>x'01' - O objecto foi pesquisado no modo binário. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito no modo binário. Se o estado de pesquisa do objecto é SCAN_FAILURE, ocorreu uma falha na pesquisa do objecto no modo binário.</p>

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZSCCSID1	INTEGER	Indica se o objecto foi pesquisado no CCSID listado, quando foi pesquisado anteriormente. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito neste CCSID. Se o estado de pesquisa do objecto é SCAN_FAILURE, ocorreu uma falha na pesquisa do objecto neste CCSID. Um valor 0 significa que este campo não é aplicável.
QEZSCCSID2	INTEGER	Indica se o objecto foi pesquisado no CCSID listado, quando foi pesquisado anteriormente. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito neste CCSID. Se o estado de pesquisa do objecto é SCAN_FAILURE, o valor deste campo será 0. Um valor 0 significa que este campo não é aplicável.
QEZUDATE	TIMESTAMP	Os segundos decorridos desde a fase correspondente à data da última utilização do objecto. O valor deste campo é zero quando o objecto é criado. Se os dados de utilização não são mantidos para o tipo OS/400 ou para o sistema de ficheiros ao qual pertence um objecto, o valor deste campo é zero.
QEZUDCOUNT	INTEGER	O número de dias de utilização de um objecto. Existem significados diferentes para a utilização de acordo com o sistema de ficheiros específico e com os tipos de objecto individuais suportados num sistema de ficheiros. A utilização pode indicar a abertura ou fecho de um ficheiro ou pode referir-se à adição de ligações, mudança de nome, restauro ou a dar saída a um objecto. Esta contagem aumenta sempre que um objecto é utilizado e volta a zero se chamar a API Qp0lSetAttr().
QEZURESET	INTEGER	Os segundos decorridos desde a última vez que a fase correspondente à contagem dos dias da data foi colocada a zero (0). Esta data é definida para a data actual quando é chamada a API Qp0lSetAttr() para repor a contagem dos Dias para zero.
QEZPRMLNK	SMALLINT	Quando um objecto tem vários nomes, este campo é definido apenas para o primeiro nome encontrado.
QEZALWCKPW	SMALLINT	Em que casos um ficheiro de dados contínuos (*STMF) pode ser partilhado com leitores e escritores durante o processamento de ponto de controlo guardar-enquanto-activo. Os valores válidos são:  0 - O objecto só pode ser partilhado com leitores.  1 - O objecto pode ser partilhado com leitores e escritores.
QEZSIG <sup>1</sup>	SMALLINT	Se um objecto tem uma assinatura digital do OS/400. Os valores válidos são:  0 - O objecto não tem uma assinatura digital OS/400.  1 - O objecto tem uma assinatura digital do OS/400.
QEZSYSSIG	SMALLINT	Se o objecto foi assinado por uma fonte considerada fidedigna pelo sistema. Os valores válidos são:  0 - Nenhuma das assinaturas é de uma fonte considerada fidedigna pelo sistema.  1 - O objecto foi assinado por uma fonte considerada fidedigna pelo sistema. Se o objecto tem várias assinaturas, pelo menos uma das assinaturas é de uma fonte considerada fidedigna pelo sistema.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZMLTSIG	SMALLINT	<p>Se um objecto tem mais do que uma assinatura digital do OS/400. Os valores válidos são:</p> <p>0 - O objecto tem apenas uma assinatura digital.</p> <p>1 - O objecto tem mais do que uma assinatura digital. Se o campo QEZSYSSIG tiver o valor 1, pelo menos uma das assinaturas é considerada fidedigna pelo sistema.</p>
QEZDSTGOPT	SMALLINT	<p>Esta opção deverá ser utilizada para determinar a atribuição de memória auxiliar pelo sistema ao objecto especificado. Esta opção só poderá ser especificada para ficheiros de dados contínuos nos sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador. Esta opção será ignorada nos ficheiros de dados contínuos de bytes *TYPE1. Os valores válidos são:</p> <p>0 - A memória auxiliar será atribuída normalmente. Ou seja, uma vez que é necessário uma memória auxiliar adicional, esta será atribuída num tamanho lógico de forma a satisfazer os requisitos de espaço actuais e futuros, ao mesmo tempo que reduz o número de operações de I/O efectuadas no disco.</p> <p>1 - A memória auxiliar será atribuída para reduzir o espaço utilizado pelo objecto. Ou seja, uma vez que é necessário uma memória auxiliar adicional, esta será atribuída com dimensões reduzidas de forma a satisfazer os requisitos de espaço actuais. O acesso a um objecto composto por muitas extensões pequenas poderá aumentar o número de operações de I/O do disco relativas a esse objecto.</p> <p>2 - O sistema determinará de forma dinâmica a atribuição de memória auxiliar ideal para o objecto, o equilíbrio de espaço utilizado versus as operações de I/O do disco. Por exemplo, se um ficheiro tiver muitas extensões pequenas, for lido e escrito com frequência, futuramente será atribuída mais memória auxiliar para reduzir o número de operações de I/O do disco. Ou, se um ficheiro for truncado com frequência, futuramente será atribuída menos memória auxiliar para reduzir o espaço utilizado. Além disso, as informações serão mantidas nos tamanhos dos ficheiros de dados contínuos do sistema e das respectivas actividades. Estas informações de tamanho dos ficheiros também serão utilizadas para ajudar a determinar a atribuição de memória auxiliar ideal para este objecto uma vez que está relacionado com outros tamanhos de objectos.</p>

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZMSTGOPT	SMALLINT	<p>Esta opção deverá ser utilizada para determinar a atribuição e utilização de memória principal pelo sistema para o objecto especificado. Esta opção só poderá ser especificada para ficheiros de dados contínuos nos sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador. Os valores válidos são:</p> <p>0 - A memória principal será atribuída normalmente. Ou seja, será atribuído e utilizado o máximo de memória principal possível. Desta forma, é reduzido o número de operações de I/O do disco, uma vez que as informações são armazenadas na cache da memória principal.</p> <p>1 - A memória principal será atribuída para reduzir o espaço utilizado pelo objecto. Ou seja, será atribuído e utilizado o mínimo de memória principal possível. Desta forma, é reduzida a utilização da memória principal, ao mesmo tempo que aumenta o número de operações de I/O do disco, uma vez que são armazenadas menos informações na cache da memória principal.</p> <p>2 - O sistema determinará de forma dinâmica a atribuição de memória principal ideal para o objecto consoante as outras actividades do sistema e conflitos da memória principal. Ou seja, se existirem poucos conflitos na memória principal, será atribuído e utilizado o máximo de memória possível de forma a reduzir o número de operações de I/O do disco. E quando existem conflitos significativos da memória principal, será atribuída e utilizada menos memória principal para reduzi-los. Esta opção só tem efeito quando a opção de paginação do conjunto de memória é *CALC. Sempre que a opção de paginação do conjunto de memória é *FIXED, o comportamento é o mesmo que STG_NORMAL. Se aceder ao objecto através de um servidor de ficheiros, esta opção não tem qualquer efeito. Em vez disso, tem um comportamento idêntico a STG_NORMAL.</p>
QEZDIRTYP2	SMALLINT	<p>O formato do objecto do directório especificado. Os valores válidos são:</p> <p>0 - O formato do directório é *TYPE1.</p> <p>1 - O formato do directório é *TYPE2.</p> <p>Para obter mais informações, consulte "Directórios *TYPE2" na página 10.</p>
QEZFILTYP2 <sup>1</sup>	SMALLINT	<p>O formato de ficheiro de dados contínuos (*STMF). Os valores válidos são:</p> <p>0 - O formato de ficheiro de dados contínuos é *TYPE1.</p> <p>1 - O formato de ficheiro de dados contínuos é *TYPE2.</p> <p>Para obter mais informações, consulte "Ficheiro de dados contínuos" na página 15.</p>
QEZUDFTYP2	SMALLINT	<p>O formato de ficheiro assumido dos ficheiros de dados contínuos (*STMF) criados no sistema de ficheiros definido pelo utilizador. Os valores válidos são:</p> <p>0 - O formato de ficheiro de dados contínuos é *TYPE1.</p> <p>1 - O formato de ficheiro de dados contínuos é *TYPE2.</p> <p>Para obter mais informações, consulte "Ficheiro de dados contínuos" na página 15.</p>

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZNONSAV	SMALLINT	Se pode ou não guardar o objecto. Os valores válidos são:  0 - O objecto será guardado.  1 - O objecto não será guardado. Além disso, se este objecto for um directório, nenhum dos objectos da subárvore do directório será guardado a menos que estejam especificados explicitamente como um objecto a guardar. A subárvore inclui todos os subdirectórios e os respectivos objectos.
QEZCLSTRSP	SMALLINT	O objecto é a memória que foi atribuída aos servidores xSeries integrados a ser utilizada como unidades de disco virtuais pelos servidores xSeries. Na perspectiva do servidor iSeries, as unidades virtuais aparecem como ficheiros de dados contínuos em bytes dentro do sistema de ficheiros integrado.  0 - O objecto não está na memória em disco virtual.  1 - O objecto está na memória em disco virtual.
QEZCASE	SMALLINT	Indica se o sistema de ficheiros que contém este objecto é sensível a maiúsculas e minúsculas.  0 - O sistema de ficheiros não é sensível a maiúsculas e minúsculas.  1 - O sistema de ficheiros é sensível a maiúsculas e minúsculas.
QEZOFLOW	SMALLINT	Indica se o objecto excedeu o conjunto de memória auxiliar em que se encontra. Os valores válidos são:  0 - O conjunto de memória auxiliar não foi excedido.  1 - O conjunto de memória auxiliar foi excedido.
QEZPCREAD	SMALLINT	Se o objecto pode ser escrito ou eliminado, se os respectivos atributos expandidos podem ser alterados ou eliminados ou o tamanho do objecto alterado. Os valores válidos são:  0 - O objecto pode ser alterado.  1 - O objecto não pode ser alterado.
QEZPCHID <sup>1</sup>	SMALLINT	Se o objecto pode ser apresentado utilizando uma listagem de directórios comum.  0 - O objecto não está oculto.  1 - O objecto está oculto.
QEZPCSYS	SMALLINT	Se o objecto é um ficheiro de sistema e é excluído nas procuras de directórios normais.  0 - O objecto não é um ficheiro de sistema.  1 - O objecto é um ficheiro de sistema.
QEZPCARC	SMALLINT	Se o objecto foi alterado desde a última verificação.  0 - O objecto não foi alterado.  1 - O objecto foi alterado.

Tabela 8. QAEZDxxxxO (Armazenar atributos de objectos) (continuação)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZSYSARC	SMALLINT	Se o objecto foi alterado e precisa de ser guardado. Está activado para quando a hora de alteração do objecto é actualizada e desactivado caso o objecto já tenha sido guardado.  0 - O objecto não foi alterado e não necessita de ser guardado. 1 - O objecto foi alterado e necessita de ser guardado.
QEZJRCVNAM	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações efectuadas ao diário no receptor de diário mais antigo. Quando o campo Aplicar informações está definido para PARTIAL_TRANSACTION o receptor de diário inicia a transacção parcial. De outro modo; o receptor de diário inicia a operação de gravação.
QEZJRCVLIB	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações efectuadas no diário ao nome da biblioteca que contém o receptor de diário.
QEZJRCVASP	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações do diário ao nome do ASP que contém o receptor de diário. Os valores válidos são:  *SYSBAS - O receptor de diário encontra-se no sistema ou no ASP do utilizador.  Dispositivo de ASP - O nome do dispositivo de ASP que contém o receptor de diário.
QEZJTRNI	GRAPHIC (1)	Este campo descreve as informações acerca do estado actual do objecto na medida em que este se encontra relacionado com limites de controlo de consolidações. Os valores válidos são:  x'00' (NONE) - Não existem transacções parciais.  x'01' (PARTIAL_TRANSACTION) - O objecto foi restaurado com as transacções parciais. Só é possível utilizar este objecto se utilizar o comando Aplicar Alterações Registadas em Diário (APYJRNCHG) ou Remover Alterações Registadas em Diário (RMVJRNCHG) para concluir ou remover as alterações efectuadas às transacções parciais.  x'02' (ROLLBACK_ENDED) - Foi concluída a operação de remoção de alterações do objecto com a opção "Terminar Remoção de Alterações" do ecrã Trabalhar com Definição de Consolidação (WRKCMTDFN). Recomenda-se que restaure o objecto uma vez que não o pode utilizar. Como última opção, é possível utilizar o comando Alterar Objecto Registado em Diário (CHGJRNOBJ) para permitir a utilização do objecto. No entanto, desta forma, o objecto pode ficar num estado inconsistente.
<p><b>Nota:</b></p> <ol style="list-style-type: none"> <li>Este campo está incluído no subconjunto de campos utilizados pelo comando PRDIRINF.</li> <li>Neste campo, só o nome do objecto é memorizado. O resto do nome do caminho é memorizado na Tabela 9 na página 79, no campo QEZDIRNAM1 se o tamanho do nome do directório estiver abaixo de 1 KB ou no QEZDIRNAM2 se os nomes dos directórios estiverem acima de 1 KB.</li> </ol>		

A Tabela 9 na página 79 é um exemplo de uma tabela que lista os directórios processados pelo comando RTVDIRINF.

Tabela 9. QAEZDxxxD (Armazenar os atributos do directório)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZDIRIDX	INTEGER	Identificador do nome do caminho (só aplicável aos directórios).
QEZDIRNAM1 <sup>1</sup>	VARGRAPHIC (1024)	O caminho do directório ascendente. Utilizado apenas quando o tamanho do caminho está abaixo de 1 KB.
QEZDIRNAM2 <sup>1</sup>	DBCLOB (16M)	O caminho do directório ascendente. Utilizado apenas quando o tamanho do caminho está acima de 1 KB de comprimento. É possível memorizar caminhos até 16 MB de comprimento.
QEZDRCCSID	INTEGER	O CCSID do directório.
QEZDREGION	GRAPHIC (2)	O ID da região do caminho do directório.
QEZLANGID	GRAPHIC (3)	O ID do idioma do caminho do directório.
QEZDIRLEN <sup>1</sup>	INTEGER	O comprimento do nome do caminho do directório.
QEZDIRFID	GRAPHIC (16)	O ID de ficheiro do directório. Um identificador associado ao objecto. É possível utilizar um ID de ficheiro com o Qp0lGetPathFromFileID() para obter um nome de caminho do objecto.
QEZDFID	INTEGER	O ID de ficheiro do directório.
QEZDIRFSID	BIGINT	O ID do sistema de ficheiros do directório.
QEZDIRGID	BIGINT	O ID de geração.
<b>Nota:</b>		
1. Este campo está incluído no subconjunto de campos utilizados pelo comando PRTDIRINF.		

A Tabela 10 é um exemplo de uma tabela criada através do comando RTVDIRINF. Esta tabela é utilizada pelo comando PRTDIRINF como forma de saber quais os ficheiros da base de dados que foram utilizados para armazenar as informações obtidas pelas diferentes instâncias do comando RTVDIRINF.

Tabela 10. QUSRSYS/QAEZDBFILE (Armazenar os ficheiros criados)

Nome do Campo	Tipo de Campo	Descrição do Campo
QEZDIRSRC	VARGRAPHIC (5000)	Caminho especificado no parâmetro DIR (RTVDIRINF).
QEZPRCCSID	INTEGER	O caminho de CCSID.
QEZPREGION	GRAPHIC (2)	O ID da região do caminho.
QEZPLANGID	GRAPHIC (3)	O ID do idioma do caminho.
QEZOBJFILE <sup>1</sup>	VARGRAPHIC (20)	O nome do ficheiro gerado para armazenar os atributos do objecto.
QEZDIRFILE <sup>1</sup>	VARGRAPHIC (20)	O nome do ficheiro gerado para armazenar os índices remissivos do directório.
QEZLIB <sup>1</sup>	VARGRAPHIC (20)	A biblioteca em que se encontram ambos os ficheiros gerados.
QEZSTRTIME	TIMESTAMP	Data/Hora de submissão do RTVDIRINF.
QEZENDTIME	TIMESTAMP	Data/Hora de conclusão do RTVDIRINF.
<b>Nota:</b>		
1. Este campo está incluído no subconjunto de campos utilizados pelo comando PRTDIRINF.		

**Aceder a dados de RTVDIRINF:** Existem várias opções para acesso aos dados das tabelas. Abaixo encontra-se uma lista com formas de acesso a dados criados através do comando RTVDIRINF:

- Utilização do comando PRTDIRINF

Este comando é utilizado para imprimir as informações do directório sobre os objectos e as informações do directório existentes no sistema de ficheiros integrado. As informações que serão impressas já se encontram armazenadas no ficheiro da base de dados especificado pelo utilizador no comando RTVDIRINF.

- Através de qualquer programa ou comando fornecido pela IBM que possa executar consultas numa tabela DB2<sup>®</sup> no iSeries.

Algumas das ferramentas mais comuns são o comando Iniciar Sessão Interactiva de SQL (STRSQL) e o iSeries Navigator.

Por exemplo, se pretende seleccionar objectos de um caminho específico (anteriormente reunidos pelo comando RTVDIRINF) que têm um tamanho de atribuição acima dos 10 KB, pode executar uma consulta como esta:

```
SELECT QEZOBJNAM, QEZALCSIZE FROM nome_da_biblioteca/QAEZDxxxxO WHERE  
QEZALCSIZE > 10240
```

- É possível criar os seus próprios programas e aceder às tabelas da base de dados através de quaisquer métodos DB válidos. Exemplos destes métodos são Instruções SQL Incorporadas e CLIs SQL.

**Utilização dos dados de RTVDIRINF:** Seguem-se alguns exemplos que mostram a importância dos dados ou a forma como pode utilizar os dados produzidos por cada uma das três tabelas.

- No caso da Tabela 8 na página 67, pode efectuar consultas para criar relatórios ou estatísticas com base em qualquer um dos campos desta tabela. O comando PRTDIRINF não inclui relatórios com base em todos os campos. Em vez disso, é utilizado um subconjunto.

- Os dados de Tabela 9 na página 79 contêm todos os directórios existentes no caminho especificado no parâmetro DIR do comando RTVDIRINF. Se pretende saber atributos específicos acerca do nome de caminho, por exemplo o CCSID, o ID do idioma ou comprimento, estes dados serão úteis. Todos os directórios armazenados nesta tabela têm também um valor exclusivo ou índice remissivo que os identifica. Na Tabela 8 na página 67, pode encontrar o mesmo campo, QEZDIRIDX, que indicará os objectos que pertencem a cada directório. Para saber quais os objectos que pertencem a um directório, pode efectuar uma consulta utilizando junções. Por exemplo, a instrução de consulta que se segue selecciona os nomes de todos os objectos existentes no directório "/MEUDIR":

```
SELECT QEZOBJNAM FROM nome_da_biblioteca/QAEZxxxx0, nome_da_biblioteca/QAEZxxxxD WHERE QEZDIRNAM1  
= "/MEUDIR" AND nome_da_biblioteca/QAEZxxxx0.QEZDIRIDX=nome_da_biblioteca/QAEZxxxxD.QEZDIRIDX
```

- A Tabela 10 na página 79 é utilizada na maior parte das vezes pelo comando PRTDIRINF para obter dados específicos sobre execuções RTVDIRINF. Exemplos desta situação são: os nomes das tabelas criadas, a biblioteca em que se encontram as tabelas e a hora de início e de fim do processamento. Poderá utilizar esta tabela para saber quando foi emitida uma RTVDIRINF ou em que tabelas procurar para efectuar uma consulta.

## Aceder utilizando APIs

Pode utilizar Interfaces de programação de aplicações (APIs) para aceder ao sistema de ficheiros integrado. Para obter mais detalhes sobre a utilização de APIs com o sistema de ficheiros integrado, consulte Executar operações utilizando APIs.

## Aceder utilizando o iSeries Navigator

O iSeries Navigator é a interface gráfica do utilizador que permite gerir e administrar os sistemas a partir do ambiente de trabalho do Windows. O iSeries Navigator torna o funcionamento e administração do seu sistema mais fáceis e mais produtivos. Por exemplo, pode copiar um perfil de utilizador para outro sistema arrastando-o de um servidor do iSeries para outro servidor do iSeries. Os assistentes dão indicações para a configuração da segurança e de serviços e aplicações de TCP/IP.

Existem muitas tarefas que pode executar utilizando o iSeries Navigator. Abaixo encontra-se uma lista de algumas tarefas comuns do sistema de ficheiros que o ajudarão a começar:

### **Trabalhar com ficheiros e pastas**

- “Criar uma pasta” na página 115
- “Remover uma pasta” na página 116
- “Dar entrada de um ficheiro” na página 115
- “Dar saída de um ficheiro” na página 115
- “Definir permissões” na página 117
- “Configurar conversão de texto do ficheiro” na página 117
- “Enviar um ficheiro ou uma pasta para outro sistema” na página 118
- “Alterar opções para uma definição do pacote” na página 118
- “Marcar uma data e uma hora para enviar o ficheiro ou a pasta” na página 118
- “Definir se os objectos devem ou não ser pesquisados” na página 119

### **Trabalhar com partilhas de ficheiros**

- “Criar uma partilha de ficheiros” na página 119
- “Alterar uma partilha de ficheiros” na página 119

### **Trabalhar com sistemas de ficheiros definidos pelo utilizador**

- “Criar um UDFS do sistema de ficheiros integrado” na página 35
- “Instalar um UDFS do sistema de ficheiros integrado” na página 36
- “Desinstalar um UDFS do sistema de ficheiros integrado” na página 36

### **Registar objectos em diário**

- “Iniciar registo em diário” na página 93
- “Terminar registo em diário” na página 93

## **Aceder utilizando o iSeries NetServer**

O Suporte do iSeries para a Vizinhança na Rede do Windows (iSeries NetServer) é uma função do IBM Operating System/400® (OS/400) que permite aos clientes com o Windows acederem a caminhos de directório e filas de output partilhados do OS/400. O iSeries NetServer permite que os PCs com o software do Windows acedam de forma contínua aos dados e impressoras que são geridos pelo seu iSeries. Os clientes de PC numa rede utilizam as funções de partilha de ficheiros e impressão incluídas nos respectivos sistemas operativos. Isto significa que não necessita de instalar software adicional no PC para poder utilizar o iSeries NetServer.

Os clientes do LINUX com o software de cliente Samba instalado também podem aceder aos dados e impressoras de forma contínua através do iSeries NetServer. Os directórios do iSeries NetServer partilhados podem ser instalados em clientes Linux como sistemas de ficheiros Samba (smbfs) de uma forma semelhante à instalação de sistemas de ficheiros de NFS que tenham sido exportados do iSeries. Para obter mais informações, consulte o tópico iSeries NetServer do Centro de Informações do iSeries.

Uma partilha de ficheiros de iSeries NetServer é um caminho de directório que o iSeries NetServer partilha com clientes na rede iSeries. Uma partilha de ficheiros pode consistir de quaisquer directórios do sistema de ficheiros integrado no iSeries. Antes de poder trabalhar com a partilha de ficheiros utilizando o iSeries NetServer, terá de criar uma partilha de ficheiros do iSeries NetServer e, se necessário, alterar uma partilha de ficheiros do iSeries NetServer utilizando o iSeries Navigator.

Para aceder a partilhas de ficheiros do sistema de ficheiros integrado utilizando o iSeries NetServer:

1. Faça clique com o botão direito do rato sobre **Iniciar** e seleccione **Explorador** para abrir o Explorador do Windows no seu PC com o Windows.
2. Abra o menu **Ferramentas** e seleccione **Definir unidade de rede**.

3. Seleccione a letra de uma unidade livre para a partilha de ficheiros (como, por exemplo, a unidade \).
4. Introduza o nome de uma partilha de ficheiros do iSeries NetServer. Por exemplo, pode introduzir a seguinte sintaxe: `\\QSYSTEM1\Nomepartilha`

**Nota:** QSYSTEM1 é o nome de sistema do iSeries NetServer e Nomepartilha é o nome da partilha de ficheiros que pretende utilizar.

5. Faça clique sobre OK.

**Nota:** Ao estabelecer ligação através do iSeries NetServer, o nome do servidor pode ser diferente do nome utilizado pelo iSeries Access Family. Por exemplo, o nome do iSeries NetServer pode ser QAS400X e o caminho para trabalhar com ficheiros pode ser `\\QAS400X\QDLS\MEUARQUIV.FLR\MEUFICH.DOC`. No entanto, o nome do iSeries Access Family pode ser AS400X e o caminho para trabalhar com ficheiros pode ser `\\AS400X\QDLS\MEUARQUIV.FLR\MEUFICH.DOC`.

Pode escolher os directórios a partilhar com a rede utilizando o iSeries NetServer. Esses directórios aparecem como primeiro nível abaixo do nome do servidor. Por exemplo, se partilhar o directório /home/rui com o nome di rru i, um utilizador poderá aceder a esse directório a partir de um PC com o nome `\\QAS400X\DIRRUI`, ou a partir de um cliente do LINUX com o nome `//qas400x/di rru i`.

O sistema de ficheiros "raiz" (/) fornece um rendimento muito mais elevado para a assistência a ficheiros de PC do que outros sistemas de ficheiros do iSeries. Pode desejar mover ficheiros para o sistema de ficheiros "raiz" (/). Consulte "Mover ficheiros ou pastas para outro sistema de ficheiros" na página 116 para obter mais informações.

Para obter mais informações acerca do iSeries NetServer e partilhas de ficheiros, consulte os seguintes tópicos da categoria **Networking** do Centro de Informações do iSeries:

- iSeries NetServer
- iSeries Partilhas de ficheiros do NetServer
- Aceder a partilhas de ficheiros do iSeries NetServer com um cliente de PC do Windows

## Aceder utilizando programas de transferência de ficheiros

O cliente do Protocolo de Transferência de Ficheiros (FTP) permite-lhe transferir ficheiros que se encontrem no seu iSeries server, incluindo os existentes nos sistemas de ficheiros "raiz" (/), QOpenSys, QSYS.LIB, de ASP independente QSYS.LIB, QOPT e QFileSvr.400. Também lhe permite transferir pastas e documentos do sistema de ficheiros dos serviços de biblioteca de documentos (QDLS). O cliente de FTP pode ser executado de forma interactiva num modo batch não assistido, em que os subcomandos do cliente são lidos de um ficheiro e as respostas a esses subcomandos são escritas num ficheiro. Também inclui outras funções para manipular ficheiros no seu servidor.

Pode utilizar o suporte de FTP para transferir ficheiros para e de qualquer um dos seguintes sistemas de ficheiros:

- Sistema de ficheiros "raiz" (/)
- Sistema de ficheiros de sistemas abertos (QOpenSys)
- Sistema de ficheiros de biblioteca (QSYS.LIB)
- Sistema de ficheiros QSYS.LIB de ASP independente
- Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)
- Sistema de ficheiros óptico (QOPT)
- Sistema de Ficheiros da Rede (NFS)
- Sistema de ficheiros NetWare (QNetWare)
- Sistema de ficheiros do iSeries NetClient (QNTC)

No entanto, tenha em consideração as seguintes restrições:

- O sistema de ficheiros integrado limita o suporte de FTP exclusivamente à transferência de dados de ficheiros. Não pode utilizar o FTP para transferir dados de atributos.
- Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente limitam o suporte do FTP a membros de ficheiro físicos, membros de ficheiro físicos origem e ficheiros de salvaguarda. Não pode utilizar o FTP para transferir outros tipos de objectos como, por exemplo, programas (\*PGM). No entanto, pode guardar outros tipos de objectos num ficheiro de salvaguarda, transferir o ficheiro de salvaguarda e, em seguida, restaurar os objectos.

Para obter informações sobre FTP, consulte os seguintes tópicos na categoria **Networking** do Centro de Informações do iSeries:

- FTP
- Transferir ficheiros com FTP

## Aceder utilizando um PC

Se o seu PC estiver ligado a um iSeries server, pode interagir com os directórios e objectos do sistema de ficheiros integrado como se estivessem armazenados no seu PC. Pode copiar objectos entre directórios utilizando a função arrastar e largar do Explorador do Windows. Conforme for necessário, pode efectivamente copiar um objecto do servidor para o PC, seleccionando o objecto na unidade do servidor e arrastando o objecto para a unidade do PC.

Quaisquer objectos copiados entre um iSeries server e PCs através da utilização da interface do Windows podem ser automaticamente convertidos entre EBCDIC e ASCII. EBCDIC significa Extended Binary-Coded Decimal Interchange Code e ASCII significa American National Standard Code for Information Interchange. O iSeries Access Family pode ser configurado para executar automaticamente esta conversão e até pode especificar que a conversão seja executada nos ficheiros que tenham uma extensão específica. Desde o OS/400 V4R4, o iSeries NetServer também pode ser configurado para executar a conversão em ficheiros.

Dependendo do tipo de objecto, pode utilizar interfaces de PC e, possivelmente, aplicações de PC para trabalhar com ele. Por exemplo, um ficheiro de dados contínuos que contém texto pode ser editado utilizando um editor de PC.

Se estiver ligado a um servidor iSeries server através do PC, o sistema de ficheiros integrado torna os directórios e objectos do servidor disponíveis para o PC. Os PCs podem trabalhar com ficheiros do sistema de ficheiros integrado através da utilização de clientes de partilha de ficheiros incorporados nos sistemas operativos Windows, de um cliente de FTP ou do iSeries Navigator (uma parte do iSeries Access Family). O seu PC utiliza clientes de partilha de ficheiros do Windows para aceder ao iSeries NetServer, que é executado no seu iSeries server.

### Transferir ficheiros utilizando FTP

O cliente de FTP permite-lhe transferir ficheiros que se encontrem no seu iSeries server, incluindo os existentes nos sistemas de ficheiros "raiz" (/), QSYS.LIB, QSYS.LIB de ASP independente, QOpenSys, QOPT e QFileSvr.400. Também lhe permite transferir pastas e documentos do sistema de ficheiros dos serviços de biblioteca de documentos (QDLS).

### Trabalhar com ficheiros utilizando o iSeries Navigator

O iSeries Access Family inclui o iSeries Navigator que é ligado a um iSeries server e torna o sistema de ficheiros integrado disponível para o PC. O iSeries Navigator é a interface gráfica do utilizador que permite gerir e administrar o seu iSeries server a partir do ambiente de trabalho do Windows.

### Trabalhar com ficheiros utilizando o iSeries NetServer

O iSeries NetServer é uma parte do OS/400 que permite que a partilha de ficheiros e impressão incorporada nos clientes do Windows funcione com o seu servidor.

**Nota:** A nova versão do iSeries Access Family baseia-se totalmente no NetServer para aceder ao sistema de ficheiros integrado. O suporte de NetServer só está disponível para ligações de TCP/IP a um iSeries server que utilize o OS/400 V4R2 e superiores.

---

## Converter directórios de \*TYPE1 para \*TYPE2

Os sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros definidos pelo utilizador (UDFS) no sistema de ficheiros integrado suportam o formato de directório \*TYPE2 como o do OS/400 V5R1. O formato de directório \*TYPE2 é um melhoramento do formato de directório \*TYPE1 original. Os directórios \*TYPE2 têm uma estrutura interna diferente dos directórios \*TYPE1 e facultam um aumento do rendimento e fiabilidade.

Pouco tempo depois de instalada a V5R3, é iniciada automaticamente a conversão para directórios \*TYPE2 de qualquer um dos sistemas de ficheiros que ainda não tenham sido convertidos para suportarem directórios \*TYPE2. Esta conversão não deverá afectar as actividades do sistema.

- "Descrição geral da conversão de \*TYPE1 para \*TYPE2"
- "Sugestões: Conversões"

## Descrição geral da conversão de \*TYPE1 para \*TYPE2

Os sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros definidos pelo utilizador (UDFS) no sistema de ficheiros integrado suportam o formato de directório \*TYPE2 como o do OS/400 V5R1. O formato de directório \*TYPE2 é um melhoramento do formato de directório \*TYPE1 original. Os directórios \*TYPE2 têm uma estrutura interna diferente dos directórios \*TYPE1 e facultam um aumento do rendimento e fiabilidade. Além do aumento de rendimento e fiabilidade, as novas funções, tais como o suporte de pesquisa do sistema de ficheiros integrado, só estão disponíveis para objectos existentes nos directórios \*TYPE2. Consulte "Suporte de pesquisa" na página 18 para obter mais informações.

Pouco tempo depois de instalada a V5R3, é iniciada automaticamente a conversão para directórios \*TYPE2 de qualquer um dos sistemas de ficheiros que ainda não tenham sido convertidos para suportarem directórios \*TYPE2. Esta conversão não deverá afectar as actividades do sistema uma vez que será executada como um trabalho em segundo plano de prioridade baixa.

Se a função de conversão ainda não tiver sido concluída e o sistema tiver um IPL normal ou anormal, a função de conversão será retomada quando for concluído o IPL. A conversão será reiniciada em todos os IPL até estarem totalmente convertidos todos os sistemas de ficheiros elegíveis.

Consulte "Determinar o estado de conversão" na página 85 para obter informações sobre a conclusão da função de conversão.

Os sistemas de ficheiros elegíveis para esta conversão automática são os sistemas de ficheiros "raiz" (/), QOpenSys e os definidos pelo utilizador para 1 a 32 ASPs. Consulte "Sugestões: ASP independente" na página 87 para obter informações sobre os sistemas de ficheiros definidos pelo utilizador em ASPs independentes.

**Nota:** Pode evitar a conversão automática para directórios \*TYPE2 se converter os sistemas de ficheiros antes da instalação da V5R3. Consulte "Utilizar directórios \*TYPE2 no OS/400 V5R1 ou V5R2" na página 11.

## Sugestões: Conversões

Pouco tempo depois de instalada a V5R3, é iniciada automaticamente a conversão para directórios \*TYPE2 de qualquer um dos sistemas de ficheiros que ainda não tenham sido convertidos para suportarem directórios \*TYPE2. Este processamento da conversão ocorrerá num modulo secundário de trabalho do sistema QFILESYS1.

Existem vários aspectos a considerar durante o processo de conversão:

- “Determinar o estado de conversão”
- “Criar perfis de utilizador”
- “Objectos com o nome mudado” na página 86
- “Considerações sobre perfis de utilizador” na página 86
- “Requisitos da memória auxiliar” na página 87
- “Sugestões: Ligação simbólica” na página 87
- “Sugestões: ASP independente” na página 87
- “Sugestões: Guardar e restaurar” na página 87
- “Regenerar Memória (RCLSTG)” na página 87
- “Pesquisa no sistema de ficheiros integrado” na página 88

### **Determinar o estado de conversão**

Pouco tempo depois de instalada a V5R3, é iniciada automaticamente a conversão para directórios \*TYPE2 de qualquer um dos sistemas de ficheiros que ainda não tenham sido convertidos para suportarem directórios \*TYPE2. Este processamento da conversão ocorrerá num modulo secundário de trabalho do sistema QFILESYS1.

Para determinar o estado de processamento da conversão, pode utilizar o comando Converter Directório (CVTDIR) da seguinte forma:

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

Esta chamada para execução do comando CVTDIR lista o formato de directório actual para os sistemas de ficheiros "raiz" (/), QOpenSys e UDFS e o sistema de ficheiros que estiver a ser convertido actualmente. Além disso, lista a prioridade actual da função de conversão, o sistema de ficheiros a ser convertido actualmente pelo sistema, o número de ligações e a percentagem de directórios processados para esse sistema de ficheiros. O sistema inicia a função de conversão com uma prioridade muito baixa (99) para que a função de conversão não afecte a actividade do sistema. No entanto, pode alterar a prioridade da função de conversão utilizando o valor \*CHGPTY para o parâmetro OPTION do comando CVTDIR. Consulte CVTDIR para obter mais informações sobre esta especificação de parâmetro.

Uma vez que o trabalho QFILESYS1 está a processar a conversão, pode visualizar o registo de trabalhos QFILESYS1 de mensagens a informar a existência de problemas na conversão. Além disso, serão enviadas várias mensagens de progresso acerca das conversões do sistema de ficheiros. Estas mensagens incluem informações como: o sistema de ficheiros a ser convertido, o número de ligações processadas no sistema de ficheiros, a percentagem de directórios processados no sistema de ficheiros, etc. Todas as mensagens de erro e muitas das mensagens de progresso são também enviadas para a fila de mensagens QSYSOPR. Deste modo, para uma consulta futura, é conveniente assegurar-se de que guarda os registos QHST ou os registos de trabalhos QFILESYS1, onde estão estas mensagens. Uma vez totalmente convertidos os sistemas de ficheiros, e se o sistema de ficheiros estiver a funcionar como previsto, pode eliminar estas informações de histórico.

### **Criar perfis de utilizador**

A função de conversão cria um perfil de utilizador que é utilizado durante a execução da função de conversão. Este perfil de utilizador tem o nome QP0FCWA. É utilizado pela função de conversão para ser proprietário dos directórios convertidos no sistema de ficheiros se não for possível ao proprietário original ter os respectivos os directórios.

Se possível, o perfil de utilizador é eliminado após a conclusão da conversão. Se for concedida a propriedade de um directório a este perfil de utilizador, é enviada a mensagem CPIA08B para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR.

## Objectos com o nome mudado

Os directórios \*TYPE2 requerem que os nomes de ligações sejam nomes UTF-16 válidos. Esta situação difere dos directórios \*TYPE1, que têm nomes de UCS2 Nível 1. Por este motivo, podem ser encontrados nomes inválidos ou duplicados durante uma conversão de directório. Quando é encontrado um nome inválido ou duplicado, ele é alterado para um nome UTF-16 exclusivo, válido e é enviada a mensagem CPIA08A para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR, apresentando o nome original e o novo nome. Caracteres combinados ou pares de caracteres substitutos inválidos contidos num nome podem causar a mudança de nome de um objecto.

Para obter mais informações sobre caracteres, consulte “Caracteres Combinados” ou “Caracteres Substitutos”

Para obter mais informações sobre UTF-16, consulte a home page Unicode (<http://www.unicode.org>



**Caracteres Combinados:** Alguns caracteres podem ser formados por mais do que um carácter Unicode. Por exemplo, existem caracteres que têm um acento ou um trema. Estes caracteres têm de ser alterados, ou normalizados, para um formato comum antes de serem armazenados no directório, de modo a que todos os objectos tenham um nome exclusivo. A normalização de um carácter combinado é um processo pelo qual o carácter é colocado num formato conhecido e previsível. O formato escolhido para os directórios \*TYPE2 é a forma canónica composta. Se existirem dois objectos num directório \*TYPE1 que contenham os mesmos caracteres combinados, eles serão normalizados para o mesmo nome. Esta situação causa uma colisão, mesmo que um objecto contenha caracteres combinados compostos e o outro contenha caracteres combinados decompostos. Deste modo, o nome de um deles será mudado antes de o objecto ser ligado ao directório \*TYPE2.

**Caracteres Substitutos:** Alguns caracteres não têm representação válida em Unicode. Estes caracteres têm alguns valores especiais; são formados por dois caracteres Unicode em dois intervalos específicos de modo a que o primeiro carácter Unicode esteja num intervalo (por exemplo, 0xD800-0xD8FF) e o segundo carácter Unicode esteja no segundo intervalo (por exemplo, 0xDC00-0xDCFF). A este par chama-se par substituto. Se um dos caracteres Unicode faltar ou estiver fora da sequência, (apenas um carácter parcial), será um nome inválido. Foram permitidos nomes deste tipo nos directórios \*TYPE1, mas não são permitidos em directórios \*TYPE2. Para que a função de conversão continue, o nome é alterado antes de o objecto ser ligado ao directório \*TYPE2, caso seja encontrado um nome que contenha um destes nomes inválidos.

## Considerações sobre perfis de utilizador

Durante a execução da conversão, são efectuadas várias tentativas para assegurar que o mesmo perfil de utilizador que é o proprietário de directórios \*TYPE1 continua a ser o proprietário dos directórios \*TYPE2 correspondentes. Uma vez que os directórios \*TYPE1 e \*TYPE2 existem momentaneamente ao mesmo tempo, essa situação afecta a quantidade de memória pertencente ao perfil de utilizador e o número de entradas no perfil de utilizador.

Os tópicos que se seguem fornecem mais informações sobre perfis de utilizador.

- “Alterar o máximo de memória para um perfil de utilizador”
- “Alterar o proprietário de um directório” na página 87

**Alterar o máximo de memória para um perfil de utilizador:** Durante o processamento da conversão de directórios, há um número de directórios que existe momentaneamente em ambos os formatos e que pertence ao mesmo perfil de utilizador. Se durante o processamento da conversão for atingido o limite máximo de memória para o perfil de utilizador, o limite do perfil de utilizador será aumentado. A mensagem CPIA08C é enviada para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR.

**Alterar o proprietário de um directório:** Se o perfil de utilizador ao qual pertence o directório \*TYPE1 não puder ser o proprietário do directório \*TYPE2 que é criado, o proprietário do directório \*TYPE2 será definido como um dos perfis de utilizador alternativos descritos em Criar perfis de utilizador. A mensagem CPIA08B é enviada para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR e a conversão continua.

### Requisitos da memória auxiliar

Devem ser considerados requisitos de memória auxiliar enquanto o sistema converte os directórios num sistema de ficheiros para o formato \*TYPE2. Existem várias questões relacionadas com requisitos de memória auxiliar:

- O tamanho final dos directórios após serem convertidos para o formato \*TYPE2
- A memória adicional necessária durante a execução da função de conversão

Em muitos casos, o tamanho final de um directório \*TYPE2 é menor do que o de um directório \*TYPE1. Normalmente, os directórios \*TYPE2 com menos de 350 objectos requerem menos memória auxiliar do que os directórios \*TYPE1 com o mesmo número de objectos. Os directórios \*TYPE2 com mais de 350 objectos são dez por cento maiores (em média) do que os directórios \*TYPE1.

Durante a execução da função de conversão, é necessária memória adicional. A função de conversão requer que os directórios tenham uma versão \*TYPE1 e uma versão \*TYPE2 simultaneamente.

**Nota:** Antes de instalar a V5R3, é conveniente executar a opção \*ESTIMATE do comando (CVTDIR) da V5R2, uma vez que poderá obter uma estimativa cautelosa da quantidade de memória auxiliar necessária durante a conversão.

### Sugestões: Ligação simbólica

Ligações simbólicas são objectos existentes no sistema de ficheiros integrado que contêm um caminho para outro objecto. Existem algumas ocorrências durante a conversão quando o nome de um objecto pode ser alterado. Se um dos elementos do caminho numa ligação simbólica sofrer uma mudança de nome durante a conversão, o conteúdo da ligação simbólica deixará de apontar para o objecto. Consulte o tópico Objectos com o nome mudado para obter detalhes sobre a mudança de nome de objectos.

### Sugestões: ASP independente

Se os sistemas de ficheiros definidos pelo utilizador num ASP independente ainda não foram convertidos para o formato de directório \*TYPE2, serão convertidos na primeira vez que o ASP independente for activado num sistema instalado com o OS/400 V5R2 ou posterior. Para fins de planeamento, é fornecida uma função de estimativa no OS/400 V5R1 para dar informações sobre o período de tempo que demorará uma conversão. Antes de activar o ASP independente para o servidor com a V5R2 ou posterior, execute a seguinte API no sistema com a V5R1 quando o ASP independente (designado por ASP\_NAME) for activado:

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

### Sugestões: Guardar e restaurar

Os directórios existentes como \*TYPE1 podem ser guardados e restaurados num sistema de ficheiros que foi convertido em \*TYPE2. Da mesma forma, os directórios existentes como \*TYPE2 podem ser guardados e restaurados num sistema de ficheiros que tenha o formato \*TYPE1, desde que não tenha sido excedido nenhum dos limites de \*TYPE1 quando o directório existia como um directório \*TYPE2.

### Regenerar Memória (RCLSTG)

Enquanto o sistema converte os sistemas de ficheiros "raiz" (/), QOpenSys e os sistemas de ficheiros de utilizador UDFS ASP para suportarem o formato de directório \*TYPE2, o comando RCLSTG não pode ser executado nos directórios do sistema de ficheiros integrado, incluindo os do ASP independente. No entanto, pode utilizar o valor de parâmetro OMIT(\*DIR) para omitir directórios do sistema de ficheiros integrado e permitir a recuperação de objectos relacionados com não directórios. Para determinar quando

os sistemas de ficheiros são totalmente convertidos para que possa executar RCLSTG, incluindo os directórios do sistema de ficheiros integrado, consulte “Determinar o estado de conversão” na página 85.

## **Pesquisa no sistema de ficheiros integrado**

Os objectos nos sistemas de ficheiros “raiz” (/), QOpenSys e sistemas de ficheiros de utilizador UDFS ASP só serão pesquisados utilizando os pontos de saída relacionados com a pesquisa, quando os sistemas de ficheiros tiverem sido totalmente convertidos para o formato de directório \*TYPE2. Os atributos relacionados com a pesquisa podem ser definidos para objectos em directórios \*TYPE1 e \*TYPE2, de modo a especificar se os objectos devem ou não ser pesquisados, mesmo que o sistema de ficheiros não esteja totalmente convertido. Para obter mais informações sobre o suporte de pesquisa do sistema de ficheiros integrado, consulte “Suporte de pesquisa” na página 18.

Enquanto o sistema converte objectos do formato de directório \*TYPE1 para o formato de directório \*TYPE2, é considerado o valor de sistema Controlo de pesquisa ‘Pesquisar no próximo acesso após restauro do objecto’, como se o objecto convertido estivesse a ser restaurado. Por exemplo, se o valor ‘Pesquisar no próximo acesso após restauro do objecto’ for especificado durante a conversão, um objecto que esteja num directório \*TYPE1 e com o atributo ‘o objecto não será pesquisado’ especificado, será pesquisado pelo menos uma vez, após a conversão total do sistema de ficheiros. Para obter mais informações sobre os valores de sistema relacionados com a pesquisa, consulte “Valores do sistema relacionados” na página 20.

Para determinar se os sistemas de ficheiros foram totalmente convertidos para que possa iniciar a pesquisa no sistema de ficheiros integrado desse sistema de ficheiros, consulte “Determinar o estado de conversão” na página 85.

---

## **Registar objectos em diário**

O principal objectivo do registo em diário é permitir recuperar as alterações efectuadas a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado. Além disso, uma das funções principais é ajudar na replicação das alterações efectuadas a um objecto para outro sistema para obter uma maior disponibilidade ou equilíbrio da carga de trabalho.

Estas informações fornecem uma breve descrição geral da gestão de registos em diário, assim como considerações para registar em diário objectos do sistema de ficheiros integrado e uma descrição de suporte de registo em diário para esses objectos.

Para obter mais informações sobre o registo de objectos em diário, consulte os tópicos que se seguem.

- “Descrição geral do registo em diário”
- “Iniciar registo em diário” na página 93
- “Alterar registo em diário” na página 93
- “Terminar registo em diário” na página 93

## **Descrição geral do registo em diário**

Os tópicos que se seguem introduzem o suporte de registo em diário para objectos do sistema de ficheiros integrado:

- “Gestão de registos em diário” na página 89
- “Objectos que deve registar em diário” na página 89
- “Objectos do sistema de ficheiros integrado registados em diário” na página 89
- “Operações registadas em diário” na página 91
- “Considerações especiais para entradas de diário” na página 91
- “Considerações sobre múltiplas ligações permanentes e o registo em diário” na página 92

Para obter informações detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte o tópico Journal management no Centro de Informações do iSeries.

## Gestão de registos em diário

O principal objectivo da gestão de registos em diário consiste em permitir que o utilizador recupere as alterações ocorridas num objecto, desde a última vez que este foi guardado. Pode também utilizar a gestão de registos em diário para:

- Um registo de auditoria da actividade que ocorre em objectos do sistema
- Registar a actividade que ocorreu nos objectos diferentes dos que podem ser registados em diário
- Uma recuperação mais rápida ao restaurar a partir do suporte de guardar enquanto activo
- Auxiliar na replicação das alterações efectuadas ao objecto para outro sistema para obter uma maior disponibilidade ou equilíbrio da carga de trabalho
- Assistência nos testes aos programas de aplicação

Pode utilizar um registo em diário para definir os objectos que pretende proteger através da gestão de registos em diário. Consulte "Objectos que deve registar em diário" para obter mais considerações acerca do registo em diário de objectos. No sistema de ficheiros integrado pode registar ficheiros de dados contínuos, directórios e ligações simbólicas em diário. Apenas são suportados os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e UDFS.

## Objectos que deve registar em diário

Ao decidir se deve registar em diário um objecto do sistema de ficheiros integrado, tenha em atenção as seguintes questões:

- Quantas vezes será o objecto alterado? Um objecto com um elevado volume de alterações entre operações de salvaguarda é uma boa opção para ser registado em diário.
- Qual a dificuldade em reconstruir as alterações efectuadas num objecto? Muitas das alterações feitas no objecto são efectuadas sem registos escritos? Por exemplo, um objecto utilizado para registo de encomendas feitas por telefone será mais difícil de reconstruir que um objecto utilizado para encomendas recebidas por correio em fichas de encomenda.
- Qual a importância das informações contidas no objecto? Caso o objecto tenha de ser restaurado até à última operação de salvaguarda, que efeito teria no negócio o atraso na reconstrução das alterações?
- Qual a relação do objecto com outros objectos no servidor? Apesar de os dados contidos num determinado objecto não serem alterados com frequência, os dados desse objecto podem ser relevantes para outros objectos mais dinâmicos do servidor. Por exemplo, muitos objectos dependem do ficheiro mestre do cliente. Caso esteja a reconstruir encomendas, o ficheiro mestre do cliente tem de incluir novos clientes ou novas alterações aos limites de crédito, que tenham sido efectuadas desde a última salvaguarda.

## Objectos do sistema de ficheiros integrado registados em diário

Alguns tipos de objecto do sistema de ficheiros integrado podem ser registados em diário através da utilização do suporte de registo em diário do OS/400. Os tipos de objecto suportados são ficheiros de dados contínuos, directórios e ligações simbólicas. Os sistemas de ficheiros "raiz" (/), QOpenSys e UDFS são os únicos que suportam o registo em diário destes tipos de objecto. Os objectos do sistema de ficheiros integrado podem ser registados em diário através da utilização de uma interface de sistema tradicional (comandos de CL ou APIs) ou da utilização do iSeries Navigator. Pode Iniciar o registo em diário e Terminar o registo em diário através do iSeries Navigator, bem como ver as informações acerca do estado do registo em diário.

**Nota:** Os ficheiros de dados contínuos definidos pela memória e os ficheiros de dados contínuos utilizados pelo Integrated xSeries Server for iSeries (IXS) para o espaço de memória de unidade virtual não podem ser registados em diário. Os directórios que podem conter objectos de ficheiro especial de blocos não podem ser registados em diário. Seguem-se alguns exemplos:  
/dev/QASP01, /dev/QASP22 e /dev/IASPNAME.

Segue-se uma lista que resume o suporte de registo em diário no sistema de ficheiros integrado:

- Pode utilizar tanto os comandos como as APIs genéricos para executar operações de registo em diário nos tipos de objecto suportados. Estas interfaces normalmente aceitam a identificação do objecto sob a forma de um nome de caminho, um ID de ficheiro ou ambos.
- Alguns comandos da operação de registo em diário, incluindo Iniciar Registo em Diário, Terminar Registo em Diário, Alterar Registo em Diário e Aplicar Alterações Registadas em Diário, podem ser executados em subárvores completas de objectos do sistema de ficheiros integrado. Como opção, pode utilizar as listas de inclusão e exclusão que podem utilizar padrões de caracteres globais para nomes de objecto. Por exemplo, pode utilizar o comando Iniciar Registo em Diário para especificar o início em todos os objectos da árvore "/MinhaEmpresa" que correspondem ao padrão "\*.data", mas excluindo quaisquer objectos que correspondam aos padrões "A\*.data" e "B\*.data".
- O suporte de registo em diário nos directórios inclui operações de directório, tais como adicionar ligações, remover ligações, criar, mudar o nome de e mover objectos dentro do directório.

Os directórios registados em diário suportam um atributo que pode ser definido para fazer com que novos objectos na subárvore herdem o actual estado de registo do directório. Quando este atributo é activado para um directório registado, o sistema inicia automaticamente o registo em diário de todos os ficheiros de dados contínuos, directórios e ligações simbólicas criados no ou ligados ao directório (através da adição de uma ligação permanente ou através da mudança do nome ou da movimentação do objecto).

**Nota:** Considerações de atributo de registo em diário herdado:

- Se mudar o nome de um objecto no mesmo directório em que reside actualmente, o registo em diário não será iniciado para o objecto, mesmo que o directório tenha o atributo de estado de registo em diário actual herdado activo.
  - Quando um directório é movido para um directório que tenha o atributo de registo em diário herdado activo, o registo em diário só será iniciado para o directório que foi movido se for apropriado. Os objectos dentro desse directório que foi movido não são afectados.
  - Se um objecto é restaurado para um directório que tenha o atributo de registo em diário herdado activo, o registo em diário do objecto não será iniciado se o mesmo nunca foi registado em diário.
  - Ao utilizar o comando Aplicar Alterações Registadas em Diário (APYJRNCHG), o valor actual do atributo de registo em diário herdado para os directórios não é utilizado. Em vez disso, é iniciado o registo em diário dos objectos criados como parte das aplicações ou que não são baseados no que aconteceu durante a actividade em tempo de execução que está a ser aplicada.
- Os nomes de objecto e os nomes completos de caminho estão contidos em várias entradas de registo em diário de objectos sistema de ficheiros integrado. Os nomes de objectos e de caminho estão activados para NLS (National Language Support).
  - Se o sistema terminar anormalmente, é fornecida uma recuperação por carregamento de programa inicial (IPL) do sistema para os objectos sistema de ficheiros integrado registados em diário.
  - O limite máximo de escrita suportado pelas várias interfaces de escrita é de 2 gigabytes-1. O tamanho máximo de uma entrada de diário se RCVSIZOPT (\*MAXOPT2 ou \*MAXOPT3) for especificado é de 4.000.000.000. Caso contrário, o tamanho máximo de uma entrada de diário é de 15. 761.440 bytes. Se registar em diário o ficheiro de dados contínuos e tiver escritas que excedam 15. 761.440 bytes, deverá utilizar o suporte \*MAXOPT2 ou \*MAXOPT3 para impedir a ocorrência de quaisquer erros.

Para obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte o tópico Journal Management.

Para obter mais informações sobre o esquema das várias entradas de diário, existe um ficheiro "include" em linguagem C, o qp0ljrn1.h, incluído no membro QSYSINC/H (QP0LJRN1), que contém detalhes dos formatos e conteúdo específicos dos dados das entradas de registo de sistema de ficheiros integrado.

Para obter uma lista completa de todas as entradas de registo depositadas para objectos do sistema de ficheiros integrado, e definições dos esquemas mencionados acima, consulte Journal entry information finder no tópico Journal Management.

### **Operações registadas em diário**

As operações que se seguem só são registadas se o tipo de objecto ou ligação que a operação está a utilizar também puder ser registado:

- Criar um objecto.
- Adicionar uma ligação a um objecto existente.
- Interromper uma ligação.
- Mudar o nome a uma ligação.
- Mudar o nome a um identificador de ficheiro.
- Mover uma ligação para ou de um directório.

As operações registadas em diário que se seguem são específicas de um ficheiro de dados contínuos:

- Escrita ou eliminação de dados
- Truncagem/extensão de ficheiros
- Dados de ficheiro forçados
- Salvaguarda com memória livre

As operações registadas em diário que se seguem aplicam-se a todos os tipos de objecto registados:

- Alterações aos atributos (incluindo alterações na segurança, tais como autoridades e propriedade)
- Abrir
- Fechar
- Iniciar registo em diário
- Comando Alterar Objecto Registado em Diário (CHGJRNOBJ)
- Terminar registo em diário
- Iniciar o comando Aplicar Alterações Registadas em Diário (APYJRNCHG)
- Terminar o comando Aplicar Alterações Registadas em Diário (APYJRNCHG)
- Guardar
- Restaurar

Para obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte o tópico Journal management. Para obter uma lista completa de todas as entradas de registo depositadas para objectos do sistema de ficheiros integrado, consulte Journal entry information finder no tópico Journal Management.

### **Considerações especiais para entradas de diário**

Muitas operações do sistema de ficheiros integrado registadas em diário utilizam internamente o controlo de consolidações para formar uma única transacção a partir de várias funções executadas durante as operações. Estas operações registadas em diário não deverão ser consideradas como completas, a menos que o ciclo do controlo de consolidações tenha uma entrada de diário de Consolidações (Código de Diário C, Tipo CM). As operações registadas que contenham uma entrada de diário Remover Alterações (Código C do Registo, Tipo RB) no ciclo de controlo de consolidação correspondem a operações falhadas e as entradas de registo incluídas nas mesmas não devem reproduzidas nem replicadas.

As entradas do sistema de ficheiros integrado registadas em diário (Código de Diário B) que utilizam o controlo de consolidações desta forma incluem:

- AA — Alterar Valor da Auditoria
- B0 — Iniciar Criação

- B1 — Criar Resumo
- B2 — Adicionar Ligação
- B3 — Mudar nome/Mover
- B4 — Desligar (Directório Ascendente)
- B5 — Interromper (Ligação)
- FA — Alteração de Atributos
- JT — Iniciar Diário (apenas quando o registo em diário é iniciado devido a uma operação num directório com o atributo de registo em diário herdado como Sim)
- OA — Alteração de Autoridades
- OG — Alteração do Grupo Primário do Objecto
- OO — Alteração do Proprietário do Objecto

Várias entradas de diário do sistema de ficheiros integrado têm um campo de dados específico que indica se a entrada é de resumo. As operações que enviam tipos de entrada de resumo enviam dois dos mesmos tipos de entrada ao registo. A primeira entrada contém um subconjunto dos dados específicos da entrada. A segunda entrada contém dados específicos da entrada completos e indicará que se trata de uma entrada de resumo. Os programas que executam a replicação do objecto ou a reprodução da operação normalmente só se interessam por entradas de resumo.

Para uma operação criar num directório registado em diário, a entrada de registo B1 (Criar Resumo) é considerada como a entrada de resumo.

Algumas operações registadas em diário precisam de enviar uma entrada de registo que esteja de modo oposto relacionada à operação. Por exemplo, um ciclo de controlo de consolidação que contém uma entrada de registo B4 (Desligar) pode também conter uma entrada de registo B2 (Adicionar Ligação). Este tipo de cenário só ocorre em operações que resultam numa entrada de diário Remover Alterações ( C — RB) .

Ele pode ocorrer por duas razões:

1. A operação estava prestes a falhar e a entrada era necessária internamente para limpeza do caminho com erro.
2. A operação foi interrompida por uma desactivação do sistema e, durante o IPL subsequente, a recuperação que tinha de enviar a entrada foi executada para remover as alterações feitas na operação interrompida.

### **Considerações sobre múltiplas ligações permanentes e o registo em diário**

Para obter informações sobre ligações permanentes, tais como, o que são e como funcionam, consulte “Ligação permanente” na página 12.

Se tiver múltiplas ligações permanentes a um objecto do sistema de ficheiros integrado registado em diário, todas as ligações deverão ser guardadas e restauradas ao mesmo tempo para que a ligação seja mantida, bem como, as informações de registo em diário associadas.

Se estiver a especificar nomes em alguns dos comandos relacionados com o registo em diário e se os nomes são na realidade múltiplas ligações permanentes, só será efectuada uma operação com o objecto. As outras ligações permanentes são essencialmente ignoradas.

Uma vez que as múltiplas ligações permanentes apontam para o mesmo objecto e a entrada de diário só tem o identificador de ficheiro (ID de Ficheiro) que é o mesmo para o objecto, quaisquer interfaces de registo em diário que apresentem o nome de caminho, por exemplo, Ver Registos em Diário (DSPJRN), apenas apresentarão um nome de ligação para o objecto. No entanto, isto não deverá causar problemas, uma vez que é possível funcionar num objecto através de qualquer nome e obter o mesmo resultado.

## Iniciar registo em diário

O principal objectivo do registo em diário é permitir recuperar as alterações efectuadas a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado.

Para iniciar o registo em diário de um objecto através do iSeries Navigator:

1. Expanda o sistema no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Faça clique com o botão direito do rato sobre o objecto que pretende registar em diário e seleccione **A Registrar em Diário...**
4. Após seleccionar as opções de registo em diário apropriadas, faça clique sobre **Iniciar**.

Para iniciar o registo em diário de um objecto através de uma interface baseada em caracteres, pode utilizar o comando Iniciar Registo em Diário (STRJRN) ou a API QjoStartJournal.

Para obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte Journal management no iSeries Information Center.

## Alterar registo em diário

O principal objectivo do registo em diário é permitir recuperar as alterações efectuadas a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado. Consulte Iniciar registo em diário para obter mais informações sobre como iniciar o registo em diário num objecto. Assim que tiver iniciado o registo em diário num objecto, sejam quais forem as razões, pode pretender alterar os atributos de registo em diário no objecto sem ter de terminar e reiniciar o registo em diário. O comando Alterar Objecto Registado em Diário (CHGJRNOBJ) permite-lhe alterar os objectos registados em diário.

## Terminar registo em diário

O principal objectivo do registo em diário é permitir recuperar as alterações efectuadas a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado. Consulte Iniciar registo em diário para obter mais informações sobre como iniciar o registo em diário num objecto. Assim que tiver iniciado o registo em diário num objecto, sejam quais forem as razões, pode pretender terminar o registo em diário neste objecto.

Para terminar o registo em diário num objecto através do iSeries Navigator:

1. Expanda o sistema no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Faça clique com o botão direito do rato sobre o objecto cujo registo em diário pretende parar e seleccione **A Registrar em Diário...**
4. Faça clique sobre **Terminar**.

Para terminar o registo em diário de um objecto através de uma interface baseada em caracteres, pode utilizar o comando Terminar Registo em Diário (ENDJRN) ou a API QjoEndJournal.

Para obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte Journal management no iSeries Information Center.

---

## Suporte de programação

A adição do sistema de ficheiros integrado ao iSeries server não afecta as aplicações existentes do servidor iSeries. As linguagens de programação, utilitários e suporte do sistema, tal como as especificações de descrição de dados, funcionam do mesmo modo que funcionavam antes da adição do sistema de ficheiros integrado.

No entanto, para tirar partido dos ficheiros de dados contínuos, directórios e outros suportes do sistema de ficheiros integrado, tem de utilizar um conjunto de interfaces de programas de aplicações (APIs) fornecido para aceder a funções do sistema de ficheiros integrado.

Para além disso, a adição do sistema de ficheiros integrado permite-lhe copiar dados entre ficheiros de base de dados físicos e ficheiros de dados contínuos. Pode executar esta cópia utilizando comandos de CL, a função de transferência de dados do iSeries Access Family ou APIs.

Os seguintes tópicos explicam como utilizar funções de cópia com ficheiros de dados contínuos de sistema de ficheiros integrado e apresentam APIs como uma forma de ter acesso a funções do sistema de ficheiros integrado:

- Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados
- Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda
- Executar operações utilizando APIs
- Suporte de sockets
- Nomenclatura e suporte internacional
- Conversão de dados
- Exemplo: Funções C do sistema de ficheiros integrado

## **Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados**

Se estiver familiarizado com as operações nos ficheiros de base de dados que utilizam as ferramentas orientadas para os registos, tais como especificações de descrição de dados (DDS), pode encontrar algumas diferenças fundamentais na utilização de ficheiros de dados contínuos. As diferenças resultam da estrutura diferente (ou talvez da falta de estrutura) dos ficheiros de dados contínuos, em comparação com os ficheiros de base de dados. Para aceder a dados num ficheiro de dados contínuos, deverá indicar uma posição de bytes e um comprimento. Para aceder a dados num ficheiro de base de dados, deverá normalmente definir os campos a serem utilizados e o número de registos a serem processados.

Devido ao facto de o utilizador predefinir o formato e as características de um ficheiro orientado para registos, o sistema operativo tem conhecimento do ficheiro e pode ajudar a evitar a execução de operações que não são apropriadas para o formato e características do ficheiro. Com os ficheiros de dados contínuos, o sistema operativo tem pouco ou nenhum conhecimento do formato do ficheiro. A aplicação deve reconhecer o ficheiro e saber como operar nele adequadamente. Os ficheiros de dados contínuos permitem um ambiente de programação extremamente flexível, mas à custa de pouca ou nenhuma ajuda fornecida pelo sistema operativo. Os ficheiros de dados contínuos são mais adequados a certas situações de programação; os ficheiros orientados para registos são mais adequados a outras situações de programação.

Existem várias maneiras de copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados no sistema de ficheiros integrado:

- Copiar dados utilizando comandos de CL
- Copiar dados utilizando APIs
- Copiar dados utilizando a função de transferência de dados

### **Copiar dados utilizando comandos de CL**

Existem dois conjuntos de comandos de CL que lhe permitem copiar dados entre ficheiros de dados contínuos e membros de ficheiros da base de dados:

- CPYTOSTMF e CPYFRMSTMF
- CPYTOIMPF e CPYFRMIMPF

### **Comandos CPYTOSTMF e CPYFRMSTMF**

Pode utilizar os comandos Copiar de Ficheiro de Dados Contínuos (CPYFRMSTMF) e Copiar para Ficheiro de Dados Contínuos (CPYTOSTMF) para copiar dados entre ficheiros de dados contínuos e membros de ficheiros de base de dados. Pode criar um ficheiro de dados contínuos a partir de um membro de ficheiro de base de dados utilizando o comando CPYTOSTMF. Também pode criar um membro de ficheiro de base de dados a partir de um ficheiro de dados contínuos utilizando o comando CPYFRMSTMF. Se o ficheiro ou membro que seja o destino da cópia não existir, será criado.

No entanto, existem algumas limitações. O ficheiro de base de dados tem de ser um ficheiro físico descrito pelo programa que contenha um único campo ou um ficheiro físico origem que contenha apenas um campo de texto. Os comandos disponibilizam uma variedade de opções para converter e formatar de novo os dados que estão a ser copiados.

Os comandos CPYTOSTMF e CPYFRMSTMF podem também ser utilizados para copiar dados entre um ficheiro de dados contínuos e um ficheiro de salvaguarda.

### Comandos CPYTOIMPF e CPYFRMIMPF

Pode também utilizar os comandos Copiar para Ficheiro de Importação (CPYTOIMPF) e Copiar do Ficheiro de Importação (CPYFRMIMPF) para copiar dados entre ficheiros de dados contínuos e membros da base de dados. Os comandos CPYTOSTMF e CPYFRMSTMF não lhe permitem mover dados de ficheiros de base de dados complexos e descritos externamente (descritos em DDS). O termo *ficheiro de importação* refere-se ao ficheiro do tipo dados contínuos; o termo refere-se normalmente a um ficheiro criado com a finalidade de copiar dados entre bases de dados heterogéneas.

Quando efectua uma cópia a partir de um ficheiro de dados contínuos (ou de importação), o comando CPYFRMIMPF permite-lhe especificar um ficheiro de definição de campo (FDF), que descreve os dados do ficheiro de dados contínuos. Também pode especificar se o ficheiro de dados contínuos é delimitado e que caracteres são utilizados para marcar os limites de cadeia, campo e registo. Também são fornecidas opções para converter tipos de dados especiais como, por exemplo, hora e data.

A conversão de dados é fornecida nestes comandos se o ficheiro de dados contínuos ou membro da base de dados destino já existir. Se o ficheiro não existir, pode utilizar o seguinte método, constituído por dois passos, para converter os dados:

1. Utilize os comandos CPYTOIMPF e CPYFRMIMPF para copiar os dados entre o ficheiro descrito externamente e um ficheiro físico origem.
2. Utilize os comandos CPYTOSTMF e CPYFRMSTMF (que permitem a conversão total dos dados, independentemente de o ficheiro destino existir ou não) para copiar entre o ficheiro físico origem e o ficheiro de dados contínuos.

Segue-se um exemplo:

```
CPYTOIMPF FROMFILE(FICHDB2) TOFILE(FICHEXP) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML('X'07') STRDLML('') DATFMT(*USA) TIMFMT(*USA)
```

O parâmetro DTAFMT especifica que o ficheiro de dados contínuos (importação) de input é delimitado; a outra opção é DTAFMT(\*FIXED), que requer a especificação de um ficheiro de definição de campo. Os parâmetros FLDDLML, RCDDLML e STRDLML identificam os caracteres que funcionam como delimitadores ou separadores dos campos, registos e cadeias.

Os parâmetros DATFMT e TIMFMT indicam o formato das informações de data e hora que são copiadas para o ficheiro de importação.

Os comandos são úteis porque podem ser colocados num programa, sendo executados inteiramente no servidor. No entanto, as interfaces são complexas.

Para obter mais informações, consulte a ajuda de comandos ou o tópico Control language (CL) do Centro de Informações do iSeries.

## Copiar dados utilizando APIs

Se pretende copiar membros de ficheiros de base de dados para um ficheiro de dados contínuos numa aplicação, pode utilizar as funções do sistema de ficheiros integrado `open()`, `read()` e `write()` para abrir um membro, ler e escrever dados nele ou noutro ficheiro. Consulte o tópico Integrated File System APIs do Centro de Informações do iSeries para obter mais informações.

## Copiar dados utilizando a função de transferência de dados

As aplicações de transferência de dados do iSeries Access Family têm a vantagem de uma interface gráfica fácil de seguir e a conversão automática de dados numéricos e de caracteres. No entanto, a transferência de dados requer a instalação do produto iSeries Access Family e a utilização de recursos de PC e do iSeries server e comunicações entre os dois.

Se tiver instalado o iSeries Access Family no PC e no seu servidor, pode utilizar as aplicações de transferência de dados para transferir dados entre ficheiros de dados contínuos e ficheiros de base de dados. Pode também transferir dados para um novo ficheiro de base de dados, com base num ficheiro de base de dados existente; para um ficheiro de base de dados descrito externamente ou para um novo ficheiro e uma nova definição de ficheiro de base de dados.

As seguintes tarefas ajudam-no a copiar e a transferir dados utilizando aplicações de transferência de dados.

- Transferir dados de um ficheiro de base de dados para um ficheiro de dados contínuos
- Transferir dados de um ficheiro de dados contínuos para um ficheiro de base de dados
- Transferir dados para uma definição de ficheiro e ficheiro de base de dados recentemente criado
- Criar um ficheiro de descrição de formatos

**Transferir dados de um ficheiro de base de dados para um ficheiro de dados contínuos:** Para transferir um ficheiro de um ficheiro de base de dados para um ficheiro de dados contínuos do servidor:

1. Estabeleça uma ligação ao servidor.
2. Defina uma unidade de rede para o caminho adequado, no sistema de ficheiros do iSeries.
3. Selecione **Transferência de Dados do iSeries server** a partir da janela iSeries Access para Windows.
4. Selecione o servidor a partir do qual pretende efectuar a transferência.
5. Selecione os nomes de ficheiros, utilizando a biblioteca e o nome de ficheiro de base de dados do iSeries como origem da cópia e a unidade de rede para a localização do ficheiro de dados contínuos resultante. Também pode escolher **Detalhes do Ficheiro de PC** para seleccionar o formato do ficheiro de PC para o ficheiro de dados contínuos. A transferência de dados suporta tipos de ficheiros de PC comuns como, por exemplo, texto ASCII, BIFF3, CSV, DIF, Texto delimitado por tabulações ou WK4.
6. Faça clique sobre **Transferir dados do iSeries** para executar a transferência de ficheiros.

Também pode executar esta movimentação de dados num trabalho batch através de aplicações de transferência de dados. Proceda do modo descrito anteriormente, mas selecione a opção de menu **Ficheiro** para guardar o pedido de transferência. A aplicação Transferência de Dados para o iSeries server cria um ficheiro `.DTT` ou `.TFR`. A aplicação Transferência de Dados do iSeries server cria um ficheiro `.DTF` ou `.TTO`. No directório iSeries Access Family, podem ser executados dois programas em batch a partir da linha de comandos:

- `RTOPCB` aceita um ficheiro `.DTF` ou `.TTO` como parâmetro
- `RFROMPCB` aceita um ficheiro `.DTT` ou `.TFR` como parâmetro

Pode definir qualquer um destes comandos para execução programada utilizando uma aplicação de marcação. Por exemplo, pode utilizar a ferramenta System Agent Tool (incluída no pacote Microsoft® Plus) para especificar o programa a executar (por exemplo, `RTOPCB MYFILE.TTO`) e a hora a que pretende executar o programa.

**Transferir dados de um ficheiro de dados contínuos para um ficheiro de base de dados:** Para transferir dados de um ficheiro de dados contínuos para um ficheiro de base de dados no servidor:

1. Estabeleça uma ligação ao servidor.
2. Defina uma unidade de rede para o caminho adequado, no sistema de ficheiros do iSeries.
3. Seleccione **Transferência de Dados para o iSeries server** a partir da janela iSeries Access para Windows.
4. Seleccione o nome do ficheiro PC que pretende transferir. Para o nome do ficheiro de PC, pode seleccionar **Procurar** para especificar a unidade de rede atribuída e escolher um ficheiro de dados contínuos. Também pode utilizar um ficheiro de dados contínuos localizado no próprio PC.
5. Seleccione o servidor onde pretende que o ficheiro de base de dados descrito externamente fique localizado.
6. Faça clique sobre **Transferir dados para o iSeries server** para executar a transferência de ficheiros.

**Nota:** Se estiver a mover dados para uma definição de ficheiro de base de dados existente no servidor, a aplicação Transferência de Dados para o iSeries server exige a utilização de um ficheiro de descrição de formato associado (FDF). Um ficheiro FDF descreve o formato de um ficheiro de dados contínuos e é criado pela aplicação Transferência de Dados do iSeries server quando os dados são transferidos de um ficheiro de base de dados para um ficheiro de dados contínuos. Para concluir a transferência de dados de um ficheiro de dados contínuos para um ficheiro de base de dados, faça clique sobre **Transferir dados para o iSeries**. Se não estiver disponível um ficheiro .FDF existente, pode rapidamente criar um ficheiro .FDF.

Também pode executar esta movimentação de dados num trabalho batch através de aplicações de transferência de dados. Proceda do modo descrito anteriormente, mas seleccione a opção de menu **Ficheiro** para guardar o pedido de transferência. A aplicação Transferência de Dados para o iSeries server cria um ficheiro .DTT ou .TFR. A aplicação Transferência de Dados do iSeries server cria um ficheiro .DTF ou .TTO. No directório IBM e(logo)server iSeries Access Family, podem ser executados dois programas em batch a partir da linha de comandos:

- RTOPCB aceita um ficheiro .DTF ou .TTO como parâmetro
- RFROMPCB aceita um ficheiro .DTT ou .TFR como parâmetro

Pode definir qualquer um destes comandos para execução programada utilizando uma aplicação de marcação. Por exemplo, pode utilizar a ferramenta System Agent Tool (incluída no pacote Microsoft Plus) para especificar o programa a executar (por exemplo, RTOPCB MYFILE.TTO) e a hora a que pretende executar o programa.

**Transferir dados para uma definição de ficheiro e ficheiro de base de dados recentemente criado:** Para transferir dados para uma definição de ficheiro e ficheiro de base de dados recentemente criado:

1. Estabeleça uma ligação ao servidor.
2. Defina uma unidade de rede para o caminho adequado, no sistema de ficheiros do iSeries.
3. Seleccione **Transferência de Dados para o iSeries server** a partir da janela iSeries Access para Windows.
4. Abra o menu **Ferramentas** da aplicação Transferência de Dados para o iSeries server.
5. Seleccione **Criar ficheiro de base de dados do iSeries**.

É apresentado um assistente que lhe permite criar um novo ficheiro de iSeries a partir de um ficheiro de PC existente. Será necessário especificar o nome do ficheiro de PC no qual se baseará o ficheiro do iSeries, o nome do ficheiro do iSeries a criar e vários outros detalhes necessários. Esta ferramenta analisa um ficheiro de dados contínuos específico para determinar o número, tipo e tamanho dos campos necessários no ficheiro de base de dados resultante. Em seguida, a ferramenta pode criar a definição do ficheiro de base de dados no servidor.

**Criar um ficheiro de descrição de formatos:** Se estiver a mover dados para uma definição de ficheiro de base de dados existente no servidor, a aplicação Transferência de Dados para o iSeries server exige a utilização de um ficheiro de descrição de formato associado (FDF). Um ficheiro FDF descreve o formato

de um ficheiro de dados contínuos e é criado pela aplicação Transferência de Dados do iSeries server quando os dados são transferidos de um ficheiro de base de dados para um ficheiro de dados contínuos.

Para criar um ficheiro .FDF:

1. Crie um ficheiro de base de dados descrito externamente com um formato que corresponda ao seu ficheiro de dados contínuos origem (número de campos, tipos de dados).
2. Crie um registo de dados temporário no ficheiro de base de dados.
3. Utilize a função Transferência de Dados do iSeries server para criar um ficheiro de dados contínuos e respectivo ficheiro .FDF associado a partir deste ficheiro de base de dados.
4. A partir deste momento, já pode utilizar a função Transferência de Dados para o iSeries server. Especifique este ficheiro .FDF com o ficheiro de dados contínuos origem que pretende transferir.

## Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda

É utilizado um ficheiro de salvaguarda com os comandos guardar e restaurar para reter dados que, de outra forma, seriam escritos em banda ou disquete. O ficheiro pode também ser utilizado como um ficheiro de base de dados para ler ou escrever registos com informações de salvaguarda/restauro. Um ficheiro de salvaguarda pode também ser utilizado para enviar objectos para outro utilizador na rede SNADS.

Pode utilizar o comando CPY para copiar um ficheiro de salvaguarda para e de um ficheiro de dados contínuos. Contudo, ao copiar um ficheiro de dados contínuos para um objecto do ficheiro de salvaguarda, os dados têm de ser dados do ficheiro de salvaguarda válidos (têm de ter origem num ficheiro de salvaguarda e de ter sido copiados para um ficheiro de dados contínuos).

Através da utilização de um cliente PC, é também possível ter acesso ao ficheiro de salvaguarda e copiar os dados para a memória do PC ou para a rede local. Não se esqueça, no entanto, que não é possível ter acesso aos dados contidos em ficheiros de salvaguarda através do Sistema de Ficheiros da Rede (NFS).

## Executar operações utilizando APIs

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado estão sob a forma de funções de linguagem C. Pode optar por um dos dois conjuntos de funções, podendo utilizar cada um deles em programas que sejam criados utilizando o Integrated Language Environment<sup>®</sup> (ILE) C/400:

- Funções C do sistema de ficheiros integrado que estão incluídas no OS/400.
- Funções C fornecidas pelo programa licenciado ILE C/400.

Para obter informações sobre os programas de saída suportados pelo sistema de ficheiros integrado, seleccione Tabela 12 na página 104.

As funções do sistema de ficheiros integrado só operam através do suporte de I/O de sequência do sistema de ficheiros integrado. São suportadas as seguintes APIs:

*Tabela 11. APIs do Sistema de Ficheiros Integrado*

Função	Descrição
access()	Determinar acessibilidade de ficheiros
accessx()	Determinar a acessibilidade de ficheiros para uma classe de utilizadores
chdir()	Mudar directório actual
chmod()	Alterar autorizações do ficheiro

Tabela 11. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
chown()	Alterar proprietário e grupo do ficheiro
close()	Fechar descritor do ficheiro
closedir()	Fechar directório
creat()	Criar novo ficheiro ou reescrever ficheiro existente
creat64()	Criar novo ficheiro ou reescrever ficheiro existente (suporte de ficheiros grandes)
DosSetFileLocks()	Bloquear e desbloquear intervalo de bytes de um ficheiro
DosSetFileLocks64()	Bloquear e desbloquear intervalo de bytes de um ficheiro (suporte de ficheiros grandes)
DosSetRelMaxFH()	Alterar o número máximo dos descritores de ficheiros
dup()	Duplicar descritor do ficheiro aberto
dup2()	Duplicar descritor do ficheiro aberto para outro descritor
faccessx()	Determinar a acessibilidade de ficheiros para uma classe de utilizadores por descritor
fchdir()	Mudar o directório actual por descritor
fchmod()	Alterar autorizações do ficheiro por descritor
fchown()	Alterar proprietário e grupo do ficheiro por descritor
fclear()	Eliminar um ficheiro
fclear64()	Eliminar um ficheiro (suporte de ficheiros grandes)
fcntl()	Executar acção de controlo de ficheiros
fpathconf()	Obter variáveis do nome de caminho configuráveis por descritor
fstat()	Obter informações sobre o ficheiro por descritor
fstat64()	Obter informações sobre o ficheiro por descritor (suporte de ficheiros grandes)
fstatvfs()	Obter informações por descritor
fstatvfs64()	Obter informações por descritor (64 bits activado)
fsync()	Sincronizar alterações ao ficheiro
ftruncate()	Truncar ficheiro
ftruncate64()	Truncar ficheiro (suporte de ficheiros grandes)
getcwd()	Obter nome de caminho do directório actual
getegid()	Obter ID do grupo efectivo
geteuid()	Obter ID do utilizador efectivo
getgid()	Obter ID de grupo real
getgrgid()	Obter informações sobre o grupo utilizando o ID do grupo
getgrnam()	Obter informações sobre o grupo utilizando o nome do grupo
getgroups()	Obter IDs do grupo
getwpsnam()	Obter informações sobre o utilizador para o nome do utilizador
getpwuid()	Obter informações sobre o utilizador para o ID do utilizador
getuid()	Obter ID de utilizador real
givedescriptor()	Dar a outro trabalho acesso ao ficheiro

Tabela 11. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
ioctl()	Executar acção de controlo de I/O do ficheiro
link()	Criar ligação ao ficheiro
lseek()	Definir posição de leitura/escrita do ficheiro
lseek64()	Definir posição de leitura/escrita do ficheiro (suporte de ficheiros grandes)
lstat()	Obter informações sobre o ficheiro ou a ligação
lstat64()	Obter informações sobre o ficheiro ou a ligação (suporte de ficheiros grandes)
mmap()	Criar uma definição de correspondências de memória
mmap64()	Criar uma definição de correspondências de memória (suporte de ficheiros grandes)
mprotect()	Alterar uma protecção da definição de correspondências de memória
msync()	Sincronizar uma definição de memória
munmap()	Remover uma definição de memória
mkdir()	Criar directório
mkfifo()	Criar ficheiro especial FIFO
open()	Abrir ficheiro
open64()	Abrir ficheiro (suporte de ficheiros grandes)
opendir()	Abrir directório
pathconf()	Obter variáveis do nome de caminho de configuração
pipe()	Criar canal de interprocessamento com sockets
pread()	Ler do descritor com deslocamento
pread64()	Ler do descritor com deslocamento (suporte de ficheiros grandes)
pwrite()	Escrever no descritor com deslocamento
pwrite64()	Escrever no descritor com deslocamento (suporte de ficheiros grandes)
QjoEndJournal()	Terminar registo em diário
QjoRetrieveJournal Information()	Obter informações do registo em diário
QJORJIDI()	Obter informações do identificador do registo em diário
QJOSJRNE()	Enviar entrada de registo em diário
QjoStartJournal()	Iniciar registo em diário
QlgAccess()	Determinar acessibilidade do ficheiro (utilizando nome de caminho activado para NLS)
QlgAccessx()	Determinar a acessibilidade de ficheiros para uma classe de utilizadores (utilizando o nome de caminho que suporta NLS)
QlgChdir()	Alterar directório actual (utilizando nome de caminho activado para NLS)
QlgChmod()	Alterar autorizações do ficheiro (utilizando nome de caminho activado para NLS)

Tabela 11. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
QlgChown()	Alterar proprietário e grupo do ficheiro (utilizando nome de caminho activado para NLS)
QlgCreat()	Criar novo ficheiro ou reescrever ficheiro existente (utilizando nome de caminho activado para NLS)
QlgCreat64()	Criar novo ficheiro ou reescrever ficheiro existente (suporte para ficheiros grandes e utilizando nome de caminho activado para NLS)
QlgCvtPathToQSYSObjName()	Processar nome de caminho do Sistema de Ficheiros Integrado para Nome do Objecto QSYS (utilizando nome de caminho activado para NLS)
QlgGetAttr()	Obter atributos do sistema para um objecto (utilizando nome de caminho activado para NLS)
QlgGetcwd()	Obter nome de caminho do directório actual (utilizando nome de caminho activado para NLS)
QlgGetPathFromFileID()	Obter nome de caminho do objecto a partir do respectivo identificador de ficheiro (utilizando nome de caminho activado para NLS)
QlgGetpwnam()	Obter informações sobre o utilizador para o nome do utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwnam_r()	Obter informações sobre o utilizador para o nome do utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwuid()	Obter informações sobre o utilizador para o ID de utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwuid_r()	Obter informações sobre o utilizador para o ID de utilizador (utilizando nome de caminho activado para NLS)
QlgLchown()	Alterar proprietário e grupo da ligação simbólica (utilizando nome de caminho activado para NLS)
QlgLink()	Criar ligação ao ficheiro (utilizando nome de caminho activado para NLS)
QlgLstat()	Obter informações sobre o ficheiro ou a ligação (utilizando nome de caminho activado para NLS)
QlgLstat64()	Obter informações sobre o ficheiro ou a ligação (suporte para ficheiros grandes e nome de caminho activado para NLS)
QlgMkdir()	Criar directório (utilizando nome de caminho activado para NLS)
QlgMkfifo()	Criar ficheiro especial FIFO (utilizando nome de caminho activado para NLS)
QlgOpen()	Abrir ficheiro (utilizando nome de caminho activado para NLS)
QlgOpen64()	Abrir ficheiro (suporte para ficheiros grandes e nome de caminho activado para NLS)
QlgOpendir()	Abrir directório (utilizando nome de caminho activado para NLS)
QlgPathconf()	Obter variáveis do nome de caminho de configuração (utilizando nome de caminho activado para NLS)
QlgProcessSubtree()	Processar directórios ou objectos numa árvore de directórios (utilizando nome de caminho activado para NLS)

Tabela 11. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
QlgReaddir()	Ler entrada do directório (utilizando nome de caminho activado para NLS)
QlgReaddir_r()	Ler entrada do directório (protecção por módulo e utilizando nome de caminho activado para NLS)
QlgReadlink()	Ler valor da ligação simbólica (utilizando nome de caminho activado para NLS)
QlgRenameKeep()	Mudar o nome do ficheiro ou directório, manter <i>novo</i> , se existir (utilizando nome de caminho activado para NLS)
QlgRenameUnlink()	Mudar o nome do ficheiro ou directório, desligar <i>novo</i> , se existir (utilizando nome de caminho activado para NLS)
QlgRmdir()	Remover directório (utilizando nome de caminho activado para NLS)
QlgSaveStgFree()	Guardar dados de objectos e libertar a respectiva memória (utilizando nome de caminho activado para NLS)
QlgSetAttr()	Definir atributos do sistema para um objecto (utilizando nome de caminho activado para NLS)
QlgStat()	Obter informações sobre o ficheiro (utilizando nome de caminho activado para NLS)
QlgStat64()	Obter informações sobre o ficheiro (suporte para ficheiros grandes e nome de caminho activado para NLS)
QlgStatvfs()	Obter informações sobre o sistema de ficheiros (utilizando nome de caminho activado para NLS)
QlgStatvfs64()	Obter informações sobre o sistema de ficheiros (suporte para ficheiros grandes e nome de caminho activado para NLS)
QlgSymlink()	Criar ligação simbólica (utilizando nome de caminho activado para NLS)
QlgUnlink()	Desligar ficheiro (utilizando nome de caminho activado para NLS)
QlgUtime()	Definir horas de acesso ao e modificação do ficheiro (utilizando nome de caminho activado para NLS)
QP0FPTOS()	Executar operações diversas em sistemas de ficheiros
QP0LCHSG()	Alterar assinatura de pesquisa
Qp0lCvtPathToSYSObjName()	Processar nome de caminho do sistema de ficheiros integrado para Nome de Objecto de QSYS
QP0LFLOP()	Executar operações diversas em objectos
Qp0lGetAttr()	Obter atributos de sistema para um objecto
Qp0lGetPathFromFileID()	Obter nome de caminho do objecto a partir do respectivo ID do ficheiro
Qp0lOpen()	Abrir ficheiro com o nome de caminho activado para NLS
Qp0lProcessSubtree()	Processar directórios ou objectos numa árvore de directórios
Qp0lRenameKeep()	Mudar o nome do ficheiro ou directório, manter <i>novo</i> , se existir
Qp0lRenameUnlink()	Mudar o nome do ficheiro ou directório, desligar <i>novo</i> se existir
QP0LROR()	Obter referências a objectos
QP0LRRO()	Obter objectos referenciados

Tabela 11. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
QP0LRTSG()	Obter assinatura de pesquisa
Qp0lSaveStgFree()	Guardar dados de objectos e libertar a respectiva memória
Qp0lSetAttr()	Definir atributos de sistema para um objecto
Qp0lUnlink()	Desligar ficheiro com o nome de caminho activado para NLS
qsyssetegid()	Definir ID de grupo efectivo
qsysseteuid()	Definir ID de utilizador efectivo
qsyssetgid()	Definir ID de grupo
qsyssetregid()	Definir IDs de grupo reais e efectivos
qsyssetreuid()	Definir IDs de utilizador reais e efectivos
qsyssetuid()	Definir ID de utilizador
QZNFRTVE()	Obter informações de exportação de NFS
read()	Ler do ficheiro
readdir()	Ler entrada de directório
readdir_r()	Ler entrada de directório (protecção por módulo)
readlink()	Ler valor da ligação simbólica
readv()	Ler do ficheiro (vector)
rename()	Mudar o nome do ficheiro ou ao directório. Pode ser definido para ter a semântica Qp0lRenameKeep() ou Qp0lRenameUnlink().
rewinddir()	Repor sequência do directório
rmdir()	Remover directório
select()	Verificar estado de I/O de vários descritores de ficheiros
stat()	Obter informações sobre o ficheiro
stat64()	Obter informações sobre o ficheiro (suporte de ficheiros grandes)
statvfs()	Obter informações sobre o sistema de ficheiros
statvfs64()	Obter informações sobre o sistema de ficheiros (suporte de ficheiros grandes)
symlink()	Criar ligação simbólica
sysconf()	Obter variáveis de configuração do sistema
takedescriptor()	Obter acesso ao ficheiro a partir de outro trabalho
umask()	Definir máscara de autorização para o trabalho
unlink()	Remover ligação ao ficheiro
utime()	Definir horas de acesso e de modificação do ficheiro
write()	Escrever no ficheiro
writev()	Escrever no ficheiro (vector)

**Nota:** Algumas destas funções também são utilizadas para sockets do OS/400 . Para obter informações sobre as restrições de utilização destas funções em determinados sistemas de ficheiros, consulte *Trabalhar com sistemas de ficheiros*. Para obter um exemplo de um programa que utiliza as funções C do sistema de ficheiros integrado, consulte “Exemplo: Funções C do Sistema de ficheiros integrado” na página 109.

Tabela 12. Programas de Saída do Sistema de Ficheiros Integrado

Função	Descrição
Pesquisa do Sistema de Ficheiros Integrado em API Fechada	Chamada durante o processamento de encerramento, tal como com a API close(). Este programa de saída tem de ser fornecido pelo utilizador.
Pesquisa do Sistema de Ficheiros Integrado em API Aberta	Chamada durante o processamento de abertura, tal como com a API open(). Este programa de saída tem de ser fornecido pelo utilizador.
Processar um Nome de Caminho	Chamado pela API Qp0lProcessSubtree() para todos os objectos na procura da API que correspondem aos critérios de selecção do programa de chamada. Este programa de saída tem de ser fornecido pelo utilizador.
Guardar Libertação de Memória	Chamado pela API Qp0lSaveStgFree() para guardar um objecto do tipo *STMF iSeries

Consulte os tópicos que se seguem para obter mais informações sobre as APIs do sistema de ficheiros integrado:

- Funções de ILE C/400
- Suporte de ficheiros grandes para APIs
- Regras de nomes de caminho para APIs
- Descritor de ficheiros
- Segurança
- Copiar dados utilizando APIs
- O tópico Application programming interfaces (APIs) do Centro de Informações do iSeries

### Funções de ILE C/400

O ILE C/400 fornece as funções C padrão definidas pelo American National Standards Institute (ANSI). Estas funções podem funcionar através do suporte de I/O da gestão de dados ou através do suporte de I/O de sequência do sistema de ficheiros integrado, dependendo do que especificar quando criar o programa em C. O compilador utilizará o I/O de gestão de dados salvo indicação em contrário.

Para indicar ao compilador que deve utilizar o I/O de sequência do sistema de ficheiros integrado, tem de especificar \*IFSIO para o parâmetro Opção de interface do sistema (SYSIFCOPT) no comando Criar Módulo de ILE C/400 (CRTCMOD) ou Criar Programa C Ligado (CRTBNDC). Se especificar \*IFSIO, as funções de I/O do sistema de ficheiros integrado são ligadas em vez das funções de I/O da gestão de dados. Com efeito, as funções C de ILE C/400 utilizam as funções do sistema de ficheiros integrado para executar I/O.

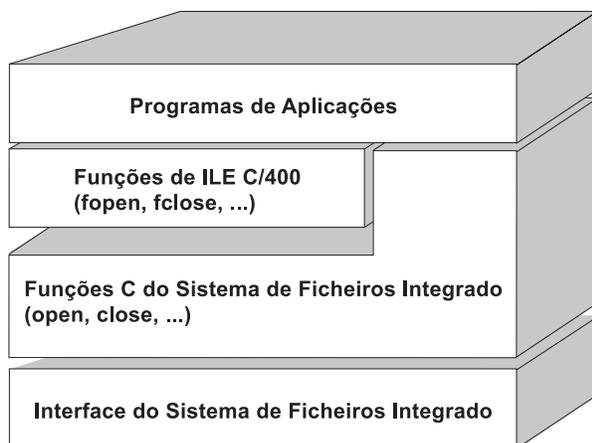


Figura 10. As funções de ILE C/400 utilizam as funções de I/O de seqüência do sistema de ficheiros integrado.

Para obter mais informações sobre a utilização de funções de ILE C/400 com I/O de seqüência do sistema de ficheiros integrado, consulte a publicação WebSphere® Development Studio: ILE C/C++

Programmers Guide . Para obter detalhes sobre cada função C de ILE C/400, consulte a publicação

WebSphere Development Studio: C/C++ Language Reference .

## Suporte de ficheiros grandes

As APIs do sistema de ficheiros integrado foram melhoradas para permitir que as suas aplicações armazenem e manipulem ficheiros muito grandes. O sistema de ficheiros integrado permite tamanhos de ficheiros de dados contínuos até aproximadamente um terabyte nos sistemas "raiz" (/), QOpenSys e nos sistemas de ficheiros definidos pelo utilizador.

O sistema de ficheiros integrado fornece um conjunto de APIs de tipo UNIX de 64 bits e permite uma definição fácil de APIs de 32 bits para APIs de 64 bits capazes de aceder a ficheiros e deslocamentos de tamanhos grandes através da utilização de argumentos de número inteiro de oito bytes. Para obter detalhes sobre cada API de 64 bits, consulte o tópico Sistema de Ficheiros Integrado APIs no Centro de Informações do iSeries.

É fornecido o seguinte para permitir às aplicações utilizar o suporte de ficheiros grandes:

1. Se a etiqueta de macro `_LARGE_FILE_API` estiver definida no momento da compilação, as aplicações terão acesso a APIs e a estruturas de dados que suportam 64 bits. Por exemplo, uma aplicação que tencione utilizar a API `stat64()` e a estrutura `stat64` terá de definir `_LARGE_FILE_API` no momento da compilação.
2. Se a etiqueta de macro `_LARGE_FILES` tiver sido definida pelas aplicações no momento da compilação, as APIs e estruturas de dados existentes serão definidas como as respectivas versões de 64 bits. Por exemplo, se uma aplicação definir `_LARGE_FILES` no momento da compilação, uma chamada para a API `stat()` será definida como a API `stat64()` e a estrutura `stat()` será definida como a estrutura `stat64()`.

A aplicação que tencionava utilizar o suporte de ficheiros grandes pode definir `_LARGE_FILE_API` no momento da compilação e serem codificadas directamente como APIs de 64 bits, mas também podem definir `_LARGE_FILES` no momento da compilação. Todas as APIs e estruturas de dados apropriadas serão, em seguida, automaticamente definidas para a versão de 64 bits.

As aplicações que não tencionam utilizar o suporte de ficheiros grandes não serão afectadas e poderão continuar a utilizar as APIs do sistema de ficheiros integrado sem alterações.

## Regras de nomes de caminho para APIs

Quando utilizar uma API do sistema de ficheiros integrado ou do ILE C/400 para trabalhar num objecto, identifique o objecto, fornecendo o respectivo caminho do directório. Nos parágrafos seguintes, poderá encontrar um resumo das regras a não esquecer quando especificar nomes de caminho nas APIs. Nestas regras, o termo **objecto** refere-se a qualquer directório, ficheiro, ligação ou outro objecto.

- Os nomes de caminho são especificados por ordem hierárquica, começando no nível mais elevado da hierarquia de directórios. O nome de cada componente no caminho é separado por uma barra (/); por exemplo:

```
Dir1/Dir2/Dir3/FichUtil
```

A barra invertida (\) não é reconhecida como um separador. É considerada como qualquer outro carácter no nome.

- Os nomes de objecto devem ser exclusivos dentro do directório.
- O comprimento máximo de cada componente do nome de caminho e o comprimento máximo da cadeia do nome de caminho podem variar em cada um dos sistemas de ficheiros. Consulte “Comparação entre sistemas de ficheiros” na página 25 para obter os limites de cada sistema de ficheiros.
- Uma / no início de um nome de caminho significa que o caminho começa no directório “raiz” (/); por exemplo:

```
/Dir1/Dir2/Dir3/FichUtil
```

- Se o nome de caminho não começar por /, assume-se que o caminho começa no directório actual; por exemplo:

```
MeuDir/MeuFich
```

onde MeuDir é um subdirectório do directório actual.

- Para evitar uma possível confusão com valores especiais do iSeries server, os nomes de caminho não podem ser iniciados por um asterisco (\*). Para especificar um nome de caminho que comece com qualquer número de caracteres, utilize dois asteriscos (\*\*); por exemplo:

```
'**.ficheiro'
```

Note que esta regra só se aplica a nomes de caminho relativos em que não existam outros caracteres antes do asterisco (\*).

- Ao trabalhar com objectos no sistema de ficheiros QSYS.LIB, os nomes de componentes têm de estar no formato *nome.tipo-objecto*; por exemplo:

```
/QSYS.LIB/SALÁRIOS.LIB/PAG.FILE
```

Consulte “Sistema de ficheiros de biblioteca (QSYS.LIB)” na página 37 para obter mais detalhes.

- Ao trabalhar com objectos no sistema de ficheiros QSYS.LIB de ASP independente, os nomes de componentes têm de estar no formato *nome.tipo-objecto*; por exemplo:

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

Consulte “QSYS.LIB de ASP independente” na página 40 para obter mais detalhes.

- Não utilize dois pontos (:) em nomes de caminho. Têm um significado especial no servidor.
- Ao contrário dos nomes de caminho nos comandos do sistema de ficheiros integrado (consulte “Regras de nomes de caminho para comandos de CL e ecrãs” na página 65), um asterisco (\*), um ponto de interrogação (?), uma plica ('), uma aspa (vírgula dobrada (")) e um til (~) não têm qualquer significado especial. São considerados como qualquer outro carácter no nome. As únicas APIs que são uma excepção a esta regra são QjoEndJournal e QjoStartJournal.
- Quando utilizar as interfaces API Qlg (utilizando nomes de caminhos activados para NLS), não é permitido um valor de carácter nulo como um dos caracteres do nome de caminho, excepto se for especificado um carácter nulo como um delimitador do nome de caminho.

## Descritor de ficheiros

Quando utilizar as funções de I/O de sequência de ILE C/400, tal como definido pelo American National Standards Institute (ANSI), para executar operações num ficheiro, identifique o ficheiro através da utilização de indicadores. Quando utilizar as funções C do sistema de ficheiros integrado, identifique o

ficheiro especificando um **descriptor de ficheiro**. Um descriptor de ficheiros é um número inteiro não negativo que deve ser único em cada trabalho. O trabalho utiliza um descriptor de ficheiros para identificar um ficheiro aberto durante a execução de operações no ficheiro. O descriptor de ficheiros é representado pela variável *files* nas funções C utilizadas no sistema de ficheiros integrado e pela variável *descriptor* nas funções C utilizadas em sockets.

Cada descriptor de ficheiros faz referência a uma **descrição de ficheiro aberto**, que contém informações, tais como posição do ficheiro, estado do ficheiro e modos de acesso ao ficheiro. A mesma descrição de ficheiro aberto pode ser referida por mais do que um descriptor do ficheiro, mas um descriptor de ficheiro pode referir-se apenas a uma descrição de ficheiro aberto.

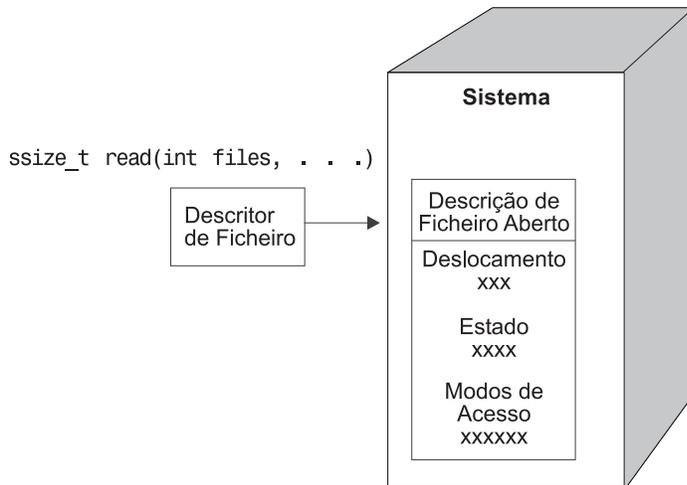


Figura 11. Descriptor de ficheiros e descrição de ficheiro aberto

Se uma função de I/O de sequência de ILE C/400 for utilizada com o sistema de ficheiros integrado, o suporte de tempo de execução de ILE C/400 converte o indicador de ficheiro num descriptor de ficheiro.

Quando utilizar os sistemas de ficheiros “raiz” (/), QOpenSys ou definidos pelo utilizador, pode transferir o acesso a uma descrição de ficheiro aberto de um trabalho para outro, permitindo, por conseguinte, que o trabalho tenha acesso ao ficheiro. Para tal, utilize a função `givedescriptor()` ou `takedescriptor()` para transferir o descriptor de ficheiros entre trabalhos. Para obter uma descrição destas funções, consulte Programação de Sockets ou o tópico Sockets APIs do Centro de Informações do iSeries.

## Segurança

Quando utilizar as APIs do sistema de ficheiros integrado, pode restringir o acesso a objectos, tal como quando utiliza interfaces de gestão de dados. No entanto, tenha em consideração que a adopção de autoridades não é suportada. Uma API do sistema de ficheiros integrado utiliza a autoridade do perfil de utilizador sob o qual o trabalho está a ser executado.

Cada um dos sistemas de ficheiros pode ter os seus próprios requisitos de autoridade especiais. Os trabalhos do servidor de NFS são a única excepção a esta regra. Os pedidos do servidor de Sistema de Ficheiros da Rede são executados no perfil do utilizador cujo número de identificação de utilizador (UID) foi recebido pelo servidor de NFS no momento do pedido.

As autoridades no servidor são o equivalente às **permissões** nos sistemas UNIX. Os tipos de permissões são leitura e escrita (para um ficheiro ou directório) e executar (para um ficheiro) ou procurar (para um directório). As permissões são indicadas por um conjunto de bits de permissão, que constituem o “modo de acesso” do ficheiro ou directório. Pode alterar os bits de permissão utilizando as funções “alterar modo” `chmod()` ou `fchmod()`. Também pode utilizar a função `umask()` para controlar quais os bits de permissão de ficheiro definidos sempre que um trabalho cria um ficheiro.

Para obter detalhes sobre segurança de dados e autoridades, consulte a publicação Security — Reference



## Suporte de sockets

Se a aplicação estiver a utilizar os sistemas de ficheiros “raiz” (/), QOpenSys ou definidos pelo utilizador, pode tirar partido do suporte de **socket local** do sistema de ficheiros integrado. Um objecto socket local (objecto do tipo \*SOCKET) permite a execução de dois trabalhos no mesmo sistema, de forma a estabelecer uma ligação de comunicações entre eles.

Um dos trabalhos estabelece um ponto de ligação utilizando a função de linguagem C `bind()` para criar o objecto socket local. O outro trabalho especifica o nome do objecto socket local na função `connect()`, `sendto()` ou `sendmsg()`. Estas funções e os conceitos gerais de sockets estão descritos no tópico Sockets Programming do Centro de Informações do iSeries.

Após estabelecer a ligação, os dois trabalhos podem enviar e receber dados entre si utilizando funções do sistema de ficheiros integrado como, por exemplo, `write()` e `read()`. De facto, nenhum dos dados que são transferidos passa, na realidade, pelo objecto socket. O objecto socket é apenas um ponto de encontro dos dois trabalhos.

Quando a comunicação entre os dois trabalhos terminar, cada trabalho utiliza a função `close()` para fechar a ligação de sockets. O objecto socket local permanece no sistema até ser removido com a utilização da função `unlink()` ou do comando Remover Ligação (RMVLNK).

O objecto socket local não pode ser guardado.

## Nomenclatura e suporte internacional

O suporte dos sistemas de ficheiros “raiz” (/) e QOpenSys garante que os caracteres nos nomes de objectos permaneçam constantes, nos diversos esquemas de codificação utilizados para diferentes idiomas nacionais e dispositivos. Quando um nome de objecto é transferido para o sistema, cada carácter do nome é convertido num formato de 16 bits, em que todos os caracteres têm uma representação codificada padrão (consulte “Continuidade de nomes” na página 17). Quando é utilizado, o nome é convertido no formato codificado apropriado para a página de códigos a ser utilizada.

Se a página de códigos para a qual o nome está a ser convertido não contiver um carácter utilizado num nome, o nome é recusado como sendo inválido.

Uma vez que os caracteres permanecem constantes entre diversas páginas de códigos, não deve efectuar uma operação pressupondo que um determinado carácter será alterado para outro carácter quando é utilizada uma página de códigos específica. Por exemplo, não deverá partir do princípio que o carácter cardinal (#) mudará para o carácter de libra esterlina, mesmo que possam ter a mesma representação codificada em páginas de códigos diferentes.

Repare que os nomes dos atributos expandidos de um objecto são convertidos do mesmo modo que o nome do objecto, por conseguinte, também se aplicam as mesmas considerações.

Para obter mais informações sobre páginas de códigos, consulte o tópico Globalization do Centro de Informações do iSeries.

## Conversão de dados

Quando acede a ficheiros através do sistema de ficheiros integrado, os dados nos ficheiros podem ou não ser convertidos, consoante o modo de abertura pedido quando o ficheiro é aberto.

Um ficheiro aberto pode estar num ou em dois modos de abertura:

## Binário

Os dados são lidos do ficheiro e escritos para o ficheiro sem conversão. A aplicação é responsável pelo tratamento dos dados.

**Texto** Os dados são lidos do ficheiro e escritos no ficheiro, assumindo que se encontram em formato de texto. Quando os dados são lidos do ficheiro, são convertidos do CCSID (coded character set identifier, identificador do conjunto de caracteres codificados) do ficheiro para o CCSID da aplicação, do trabalho ou do sistema que recebe os dados. Quando os dados são escritos no ficheiro, são convertidos do CCSID da aplicação, do trabalho ou do sistema para o CCSID do ficheiro. Para os verdadeiros ficheiros de dados contínuos, quaisquer caracteres de formatação de linhas (tais como mudança de linha, tabulação e fim de ficheiro) são apenas convertidos de um CCSID para outro.

Quando está a ler a partir de ficheiros de registos que estão a ser utilizados como ficheiros de dados contínuos, os caracteres de fim de linha (mudança de linha) são acrescentados ao fim dos dados em cada registo. Quando o sistema escreve para ficheiros de registo:

- Os caracteres de fim de linha são removidos.
- Os caracteres de tabulação são substituídos pelo número apropriado de espaços em branco até à posição de tabulação seguinte.
- As linhas são preenchidas quer com espaços em branco (para um membro de ficheiro físico origem) quer com nulos (para um membro de ficheiro físico de dados) até ao fim do registo.

Num pedido de abertura, pode especificar uma das seguintes opções:

### Binário, Forçado

Os dados são processados como binários, independentemente do conteúdo real do ficheiro. A aplicação é responsável pelo processo de tratamento dos dados.

### Texto, Forçado

Assume-se que os dados são texto. Os dados são convertidos do CCSID do ficheiro para o CCSID da aplicação.

É utilizado o valor assumido *Binário, Forçado* para a função do sistema de ficheiros integrado `open()`.

## Exemplo: Funções C do Sistema de ficheiros integrado

Este simples programa em linguagem C ilustra a utilização de várias funções do sistema de ficheiros integrado. O programa executa as seguintes operações:

- 1** Utiliza a função `getuid()` para determinar o ID de utilizador (uid) real.
- 2** Utiliza a função `getcwd()` para determinar o directório actual.
- 3** Utiliza a função `open()` para criar um ficheiro. Concede a autoridade de leitura, escrita e execução do ficheiro ao proprietário (a pessoa que criou o ficheiro).
- 4** Utiliza a função `write()` para escrever uma cadeia de bytes no ficheiro. O descritor de ficheiro que foi fornecido na operação de abertura (**3**) identifica o ficheiro.
- 5** Utiliza a função `close()` para fechar o ficheiro.
- 6** Utiliza a função `mkdir()` para criar um novo subdirectório no directório actual. É concedido ao proprietário acesso de leitura, escrita e execução no subdirectório.
- 7** Utiliza a função `chdir()` para passar do novo subdirectório para o directório actual.
- 8** Utiliza a função `link()` para criar uma ligação ao ficheiro anteriormente criado (**3**).
- 9** Utiliza a função `open()` para abrir o ficheiro apenas para leitura. A ligação que foi criada em (**8**) permite o acesso ao ficheiro.
- 10** Utiliza a função `read()` para ler uma cadeia de bytes a partir do ficheiro. O descritor de ficheiro que foi fornecido na operação de abertura (**9**) identifica o ficheiro.

- 11** Utiliza a função **close()** para fechar o ficheiro.
- 12** Utiliza a função **unlink()** para remover a ligação ao ficheiro.
- 13** Utiliza a função **chdir()** para passar do directório actual novamente para o directório ascendente, onde o novo subdirectório foi criado.
- 14** Utiliza a função **rmdir()** para remover o subdirectório criado anteriormente ( **6** ).
- 15** Utiliza a função **unlink()** para remover o ficheiro criado anteriormente ( **3** ).

**Nota:** Este programa exemplo será correctamente executado nos sistemas em que o CCSID do trabalho que é executado seja 37. As APIs do sistema de ficheiros integrado têm de ter os nomes de objectos e de caminhos codificados no CCSID do trabalho; no entanto, o compilador de C armazena constantes de caracteres no CCSID 37. Para compatibilidade total, converta as constantes de caracteres, como por exemplo, nomes de objectos e caminhos, antes de transferir APIs para o CCSID do trabalho.

```
#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        "Olá Mundo!"
#define USER_ID          "id_util_"
#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("Foi encontrado um erro, a limpar.\n");
    switch ( level )
    {
        case 1:
            printf("Não foi possível determinar directório de trabalho actual.\n");
            break;
        case 2:
            printf("Não foi possível criar ficheiro %s.\n",TEST_FILE);
            break;
        case 3:
            printf("Não foi possível gravar no ficheiro %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 4:
            printf("Não foi possível fechar ficheiro %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 5:
            printf("Não foi possível criar directório %s.\n",NEW_DIRECTORY);
```

```

        unlink(TEST_FILE);
        break;
    case 6:
        printf("Não foi possível passar para o directório %s.\n", NEW_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 7:
        printf("Não foi possível criar ligação %s a %s.\n", LinkName,InitialFile);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 8:
        printf("Não foi possível abrir ligação %s.\n",LinkName);
        unlink(LinkName);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 9:
        printf("Não foi possível ler ligação %s.\n",LinkName);
        close(FilDes);
        unlink(LinkName);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 10:
        printf("Não foi possível fechar ligação %s.\n",LinkName);
        close(FilDes);
        unlink(LinkName);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 11:
        printf("Não foi possível remover ligação %s.\n",LinkName);
        unlink(LinkName);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 12:
        printf("Não foi possível passar para o directório %s.\n",
PARENT_DIRECTORY);
        chdir(PARENT_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 13:
        printf("Não foi possível remover directório %s.\n",NEW_DIRECTORY);
        rmdir(NEW_DIRECTORY);
        unlink(TEST_FILE);
        break;
    case 14:
        printf("Não foi possível remover ficheiro %s.\n",TEST_FILE);
        unlink(TEST_FILE);
        break;
    default:
        break;
}
printf("O Programa terminou com Erros.\n"\
"É possível que nem todos os ficheiros e directórios de teste tenham sido removidos.\n");
}

int main ()

```

```

{
1
/* Obtém e imprime o verdadeiro ID do utilizador com a função getuid(). */
  UserID = getuid();
  printf("O verdadeiro ID de utilizador é %u.
\n",UserID);

2
/* Obtém o directório de trabalho actual e guarda-o em InitialDirectory. */
  if ( NULL == getcwd(InitialDirectory,BUFFER_SIZE) )
  {
    perror("Erro na função getcwd");
    CleanupOnError(1);
    return 0;
  }
  printf("O directório de trabalho actual é %s. \n",InitialDirectory);

3
/* Cria o ficheiro TEST_FILE para escrita, se não existir.
Concede ao proprietário autoridade para ler, escrever e executar. */
  FilDes = open(TEST_FILE, O_WRONLY | O_CREAT | O_EXCL, S_IRWXU);
  if ( -1 == FilDes )
  {
    perror("Erro na função open");
    CleanupOnError(2);
    return 0;
  }
  printf("Foi criado %s no directório %s.\n",TEST_FILE,InitialDirectory);

4
/* Escrever TEST_DATA para TEST_FILE via FilDes */
  BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
  if ( -1 == BytesWritten )
  {
    perror("Erro na função write");
    CleanupOnError(3);
    return 0;
  }
  printf("Foi escrito %s no ficheiro %s.\n",TEST_DATA,TEST_FILE);

5
/* Fecha TEST_FILE via FilDes */
  if ( -1 == close(FilDes) )
  {
    perror("Erro na função close");
    CleanupOnError(4);
    return 0;
  }
  FilDes = -1;
  printf("O ficheiro %s está fechado.\n",TEST_FILE);

6
/* Cria um novo directório no directório de trabalho actual e
concede ao proprietário autoridade de leitura, escrita e execução */
  if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )
  {
    perror("Erro na função mkdir");
    CleanupOnError(5);
    return 0;
  }
  printf("Foi criado o directório %s no directório %s.\n",NEW_DIRECTORY,InitialDirectory);

7
/* Passa do directório de trabalho actual para o
directório NEW_DIRECTORY criado. */
  if ( -1 == chdir(NEW_DIRECTORY) )
  {

```

```

        perror("Erro na função chdir");
        CleanupOnError(6);
        return 0;
    }
    printf("Passagem para o directório %s/%s.\n",
InitialDirectory,NEW_DIRECTORY);

/* Copia PARENT_DIRECTORY para InitialFile e
acrescenta "/" e TEST_FILE a InitialFile. */
strcpy(InitialFile,PARENT_DIRECTORY);
strcat(InitialFile,"/");
strcat(InitialFile,TEST_FILE);

/* Copia USER_ID para LinkName e, depois, acrescenta o
ID de utilizador como uma cadeia a LinkName. */
strcpy(LinkName, USER_ID);
sprintf(Buffer, "%d\0", (int)UserID);
strcat(LinkName, Buffer);

8
/* Cria uma ligação ao nome InitialFile com o LinkName. */
if ( -1 == link(InitialFile,LinkName) )
{
    perror("Erro na função link");
    CleanupOnError(7);
    return 0;
}
printf("Foi criada uma ligação %s a %s.\n",LinkName,InitialFile);

9
/* Abre o ficheiro LinkName apenas para leitura. */
if ( -1 == (FilDes = open(LinkName,O_RDONLY)) )
{
    perror("Erro na função open");
    CleanupOnError(8);
    return 0;
}
printf("%s está aberto para leitura.\n",LinkName);

10
/* Lê do ficheiro LinkName, via FilDes, para Buffer. */
BytesRead = read(FilDes,Buffer,sizeof(Buffer));
if ( -1 == BytesRead )
{
    perror("Erro na função read");
    CleanupOnError(9);
    return 0;
}
printf("%s foi lido a partir de %s.\n",Buffer,LinkName);
if ( BytesRead != BytesWritten )
{
    printf("AVISO: o número de bytes lidos não "\
"é igual ao número de bytes escritos.\n");
}

11
/* Fecha o ficheiro LinkName via FilDes. */
if ( -1 == close(FilDes) )
{
    perror("Erro na função close");
    CleanupOnError(10);
    return 0;
}
FilDes = -1;
printf("%s fechado.\n",LinkName);

12

```

```

/* Termina a ligação LinkName a InitialFile. */
if ( -1 == unlink(LinkName) )
{
    perror("Erro na função unlink");
    CleanupOnError(11);
    return 0;
}
printf("%s está removido.\n",LinkName);

13
/* Passa do directório de trabalho actual
de novo para o directório inicial. */
if ( -1 == chdir(PARENT_DIRECTORY) )
{
    perror("Erro na função chdir");
    CleanupOnError(12);
    return 0;
}
printf("A passar o directório para %s.\n",InitialDirectory);

14
/* Remove o directório NEW_DIRECTORY */
if ( -1 == rmdir(NEW_DIRECTORY) )
{
    perror("Erro na função rmdir");
    CleanupOnError(13);
    return 0;
}
printf("A remover o directório %s.\n",NEW_DIRECTORY);

15
/* Desliga o ficheiro TEST_FILE */
if ( -1 == unlink(TEST_FILE) )
{
    perror("Erro na função unlink");
    CleanupOnError(14);
    return 0;
}
printf("A remover o ficheiro %s.\n",TEST_FILE);

printf("O programa foi concluído com êxito.\n");
return 0;
}

```

**Nota:** Consulte “Informações de exclusão de responsabilidade para códigos” na página 2 para obter informações legais importantes.

---

## Trabalhar com ficheiros e pastas através da utilização do iSeries Navigator

Pode executar as seguintes tarefas com ficheiros e pastas:

- “Dar entrada de um ficheiro” na página 115
- “Dar saída de um ficheiro” na página 115
- “Criar uma pasta” na página 115
- “Remover uma pasta” na página 116
- “Mover ficheiros ou pastas para outro sistema de ficheiros” na página 116
- “Definir permissões” na página 117
- “Configurar conversão de texto do ficheiro” na página 117
- “Enviar um ficheiro ou uma pasta para outro sistema” na página 118

- “Alterar opções para uma definição do pacote” na página 118
- “Marcar uma data e uma hora para enviar o ficheiro ou a pasta” na página 118
- “Criar uma partilha de ficheiros” na página 119
- “Alterar uma partilha de ficheiros” na página 119
- “Definir se os objectos devem ou não ser pesquisados” na página 119

## Dar entrada de um ficheiro

Para dar entrada de um ficheiro:

1. No **iSeries Navigator**, faça clique com o botão direito do rato sobre o ficheiro de que pretende dar entrada.
2. Seleccione **Propriedades**.
3. Seleccione **Propriedades do Ficheiro -> Utilizar Página**.
4. Faça clique sobre **Dar Entrada**.

Caso prefira utilizar a função arrastar, pode instalar um UDFS arrastando-o para uma pasta dentro do sistema de ficheiros integrado, no mesmo servidor. Não pode largar o UDFS no /dev, /dev/QASPxx, /dev/nome\_asp, noutra sistema, nem no ambiente de trabalho.

## Dar saída de um ficheiro

Para dar saída de um ficheiro:

1. No **iSeries Navigator**, faça clique com o botão direito do rato sobre o ficheiro de que pretende dar saída.
2. Seleccione **Propriedades**.
3. Seleccione **Propriedades do Ficheiro -> Utilizar Página**.
4. Faça clique sobre **Dar Saída**.

## Criar uma pasta

Para criar uma pasta:

1. Expanda o sistema que pretende utilizar no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Faça clique com o botão direito do rato sobre o sistema de ficheiros ao qual pretende adicionar a nova pasta e seleccione **Nova Pasta**.
5. Escreva um novo nome para o objecto na caixa de diálogo **Nova Pasta**.
6. Faça clique sobre **OK**.

Convém também que especifique se os objectos criados nesta pasta devem ou não ser pesquisados. Para obter mais informações sobre este tópico, consulte “Definir se os objectos devem ou não ser pesquisados” na página 119.

Quando cria uma pasta no servidor do iSeries, tem de considerar se pretende proteger a nova pasta (ou objecto) com a gestão de registos em diário. Consulte o tópico Gestão de registos em diário para obter mais informações.

Tópicos relacionados:

- Iniciar registo em diário
- Terminar registo em diário

## Remover uma pasta

Para remover uma pasta:

1. Expanda o sistema que pretende utilizar no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**. Continue a expandir até o ficheiro ou a pasta que pretende remover estar visível.
4. Faça clique com o botão direito do rato sobre o ficheiro ou a pasta e seleccione **Eliminar**.

## Mover ficheiros ou pastas para outro sistema de ficheiros

Cada sistema de ficheiros tem as suas próprias características. No entanto, mover objectos para um sistema de ficheiros diferente pode significar a perda das vantagens do sistema de ficheiros em que os objectos estão armazenados actualmente. É conveniente mover objectos de um sistema de ficheiros para outro para beneficiar dessas características. Antes de mover objectos para outro sistema de ficheiros, deverá familiarizar-se com os sistemas de ficheiros no sistema de ficheiros integrado e respectivas características. Para obter mais informações, consulte *Trabalhar com sistemas de ficheiros*.

Também deverá considerar os seguintes aspectos:

- Está a utilizar aplicações que utilizam as vantagens do sistema de ficheiros onde os objectos se encontram?

Alguns sistemas de ficheiros suportam interfaces que não fazem parte do suporte de sistema de ficheiros integrado. As aplicações que utilizam estas interfaces poderão já não aceder a objectos que foram movidos para outro sistema de ficheiros. Por exemplo, os sistemas de ficheiros QDLS e QOPT suportam o sistema de ficheiros hierárquico (HFS). As APIs e os comandos trabalham com objectos de documentos e pasta. Não pode utilizar estas interfaces em objectos que se encontram noutros sistemas de ficheiros.

- Que características dos objectos são importantes para si?

Nem todas as características são suportadas por todos os sistemas de ficheiros. Por exemplo, os sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente suportam o armazenamento e recuperação de apenas alguns atributos expandidos, enquanto os sistemas de ficheiros "raiz" (/) e QOpenSys suportam o armazenamento e a recuperação de todos os atributos expandidos. Assim, o QSYS.LIB e o QSYS.LIB de ASP independente não são bons candidatos para o armazenamento de objectos que tenham atributos expandidos.

Os ficheiros de PC armazenados no QDLS são bons candidatos para serem movidos. A maioria das aplicações de PC deverá poder continuar a trabalhar com os ficheiros de PC que forem movidos do QDLS para outros sistemas de ficheiros. Os sistemas de ficheiros "raiz" (/), QOpenSys, QNetWare e QNTC são boas opções para armazenar estes ficheiros de PC. Como suportam muitas das características do sistema de ficheiros OS/2, estes sistemas de ficheiros podem fornecer um acesso mais rápido a ficheiros. Consulte "Comparação entre sistemas de ficheiros" na página 25 para ver uma comparação das diferentes características de cada sistema de ficheiros.

Para mover objectos para outro sistema de ficheiros, execute os seguintes passos:

1. Guarde uma cópia de todos os objectos que tenciona mover.

A criação de uma cópia de segurança permite-lhe restaurar os objectos para o sistema de ficheiros original, caso verifique que as aplicações não conseguem aceder aos objectos no sistema de ficheiros para o qual foram movidos.

**Nota:** Não pode guardar objectos de um sistema de ficheiros e restaurá-los para outro.

2. Crie os directórios no sistema de ficheiros para o qual pretende mover os objectos, utilizando o comando Criar Directório (CRTDIR).

Deve examinar com cuidado os atributos do directório onde os objectos se encontram actualmente para determinar se deseja duplicar esses atributos nos directórios que criar. Por exemplo, o utilizador que cria o directório é o respectivo proprietário, e não o utilizador proprietário do directório antigo. É

conveniente transferir a propriedade do directório depois de o ter criado, se o sistema de ficheiros suportar a definição do proprietário de um directório.

3. Mova os ficheiros para o sistema de ficheiros seleccionado, utilizando o comando Mover (MOV).

Recomenda-se a utilização do comando MOV porque mantém a propriedade dos objectos, se o sistema de ficheiros suportar a definição da propriedade dos objectos. Pode, contudo, utilizar o comando Copiar (CPY) para preservar a propriedade dos objectos, utilizando o parâmetro OWNER(\*KEEP). Não se esqueça que este procedimento só funciona em sistemas de ficheiros que suportem a definição do proprietário de um objecto. Repare que, quando utilizar MOV ou CPY:

- Os atributos podem não corresponder, pelo que poderão ser eliminados.
- Os atributos expandidos poderão ser eliminados.
- As autoridades podem não ser equivalentes, pelo que poderão ser eliminadas.

Isto significa que, se decidir devolver o objecto ao sistema de ficheiros de origem, pode não querer movê-lo ou copiá-lo de novo, devido aos atributos e autoridades eliminadas. A forma mais segura de devolver um objecto é restaurar uma versão guardada do mesmo.

## Definir permissões

A adição de permissões a um objecto permite controlar a capacidade de manipulação por terceiros desse objecto. Com as permissões, pode permitir que alguns utilizadores só tenham capacidade de visualização dos objectos, enquanto permite que outros possam de facto editá-los.

Para definir permissões para um ficheiro ou uma pasta:

1. Expanda o sistema que pretende utilizar na janela **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**. Continue a expandir até o objecto para o qual pretende adicionar permissões estar visível.
4. Faça clique com o botão direito do rato sobre o objecto para o qual pretende adicionar permissões e seleccione **Permissões**.
5. Faça clique sobre **Adicionar** na caixa de diálogo **Permissões**.
6. Seleccione um ou mais utilizadores e grupos ou introduza o nome de um utilizador ou de um grupo no campo de nomes de utilizadores ou grupos, na caixa de diálogo **Adicionar**.
7. Faça clique sobre **OK**. Esta acção adiciona os utilizadores ou os grupos ao topo da lista.
8. Faça clique no botão **Detalhes** para implementar permissões detalhadas.
9. Aplique as permissões desejadas para o utilizador, seleccionando a caixa do quadrado de opção adequado.
10. Faça clique sobre **OK**.

## Configurar conversão de texto do ficheiro

Pode configurar a conversão automática de ficheiros de texto no iSeries Navigator. A conversão automática de ficheiros de texto permite-lhe utilizar extensões de ficheiro para a conversão de dados de ficheiro. O sistema de ficheiros integrado pode converter um ficheiro de dados quando é transferido entre um iSeries e um PC. Quando obtém acesso ao ficheiro de dados a partir de um PC, ele é tratado como se estivesse em ASCII.

Para configurar conversão de texto do ficheiro:

1. Expanda o sistema que pretende utilizar no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Faça clique com o botão direito do rato sobre **Sistema de Ficheiros Integrado** e seleccione **Propriedades**.
4. Introduza a extensão de ficheiro que pretende converter automaticamente na caixa de texto **Extensões de ficheiros para conversão automática de ficheiros de texto** e faça clique sobre **Adicionar**.

5. Repita o passo 4 para todas as extensões de ficheiro que pretende converter automaticamente.
6. Faça clique sobre **OK**.

## Enviar um ficheiro ou uma pasta para outro sistema

Para enviar um ficheiro ou uma pasta para outro sistema:

1. Expanda o sistema que pretende utilizar no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**. Continue a expandir até o ficheiro ou a pasta que pretende enviar estar visível.
4. Faça clique com o botão direito do rato sobre o ficheiro ou a pasta e seleccione **Enviar**. O ficheiro ou a pasta aparece na lista **Ficheiros e Pastas Seleccionados**, na caixa de diálogo **Enviar Ficheiros de**.
5. Expanda a lista de sistemas e grupos disponíveis.
6. Seleccione um sistema e faça clique sobre **Adicionar**, para adicionar o sistema à lista **Sistemas e grupos destino**. Repita este passo para todos os sistemas para os quais pretenda enviar este ficheiro ou pasta.
7. Faça clique sobre **OK** para enviar o ficheiro ou a pasta com as actuais definições do pacote e informações da marcação assumidas.

Também pode “Alterar opções para uma definição do pacote” ou “Marcar uma data e uma hora para enviar o ficheiro ou a pasta”.

Quando cria uma definição de pacote, ela é guardada e pode ser reutilizada em qualquer momento para enviar o conjunto de ficheiros e pastas definidos para vários sistemas ou grupos de sistemas de ponto de terminação. Se optar por criar um instantâneo dos seus ficheiros, pode manter mais do que uma versão das cópias do mesmo conjunto de ficheiros. O envio do instantâneo assegura que não serão efectuadas alterações aos ficheiros durante a distribuição, de modo a que o último sistema destino receba os mesmos objectos que o primeiro sistema destino.

## Alterar opções para uma definição do pacote

Uma definição do pacote permite-lhe agrupar um conjunto de objectos ou ficheiros do sistema de ficheiros integrado do OS/400. A definição de pacote também lhe permite visualizar este mesmo grupo de ficheiros como um conjunto lógico, ou como um conjunto físico, tirando um instantâneo dos ficheiros de modo a preservá-los para distribuição posterior.

Para alterar as opções para as definições do pacote:

1. Execute os passos incluídos em “Enviar um ficheiro ou uma pasta para outro sistema”.
2. Faça clique sobre o separador **Opções**. As definições assumidas são incluir subpastas ao condensar e enviar ficheiros e substituir um ficheiro existente pelo enviado.
3. Altere estas opções se necessário.
4. Faça clique sobre **Avançada** para definir as opções de salvaguarda e restauro avançadas.
5. Faça clique sobre **OK** para guardar as opções avançadas.
6. Faça clique sobre **OK** para enviar o ficheiro ou faça clique sobre **Marcar**, para definir uma hora para enviar o ficheiro.

Tópicos relacionados:

- “Marcar uma data e uma hora para enviar o ficheiro ou a pasta”.

## Marcar uma data e uma hora para enviar o ficheiro ou a pasta

A utilização da função do marcador dá-lhe flexibilidade para executar o seu trabalho quando for conveniente para si. Para marcar uma data e uma hora para enviar o ficheiro ou a pasta:

1. Execute os passos incluídos em “Enviar um ficheiro ou uma pasta para outro sistema” na página 118.
2. Faça clique sobre **Marcar**.
3. Selecciona as opções relativas à altura em que pretende enviar o ficheiro ou a pasta.

## Criar uma partilha de ficheiros

Uma partilha de ficheiros é um caminho de directório que o iSeries NetServer partilha com clientes de PC na rede do iSeries. Uma partilha de ficheiros pode consistir de quaisquer directórios do sistema de ficheiros integrado no iSeries.

Para criar uma partilha de ficheiros:

1. Expanda o sistema em **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda o sistema de ficheiros que contém a pasta para o qual pretende criar uma partilha.
5. Faça clique com o botão direito do rato sobre a pasta para a qual pretende criar uma partilha e seleccione **Partilhar**.
6. Selecciona **Nova Partilha**.

## Alterar uma partilha de ficheiros

Uma partilha de ficheiros é um caminho de directório que o iSeries NetServer partilha com clientes de PC na rede do iSeries. Uma partilha de ficheiros pode consistir de quaisquer directórios do sistema de ficheiros integrado no iSeries.

Para alterar uma partilha de ficheiros:

1. Expanda o sistema em **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda a pasta com a partilha definida que pretende alterar.
5. Faça clique com o botão direito do rato sobre a pasta com a partilha definida que pretende **Partilhar**.
6. Selecciona **Nova Partilha**.

## Definir se os objectos devem ou não ser pesquisados

Para definir se um objecto deve ou não ser pesquisado:

1. Expanda o sistema em **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda a pasta ou o ficheiro pretendido.
5. Faça clique com o botão direito do rato sobre a pasta ou o ficheiro e seleccione **Propriedades**.
6. Selecciona o separador **Segurança**.
7. Selecciona **Pesquisar objectos** com a opção pretendida.

Para obter mais informações sobre as opções, consulte abaixo. As descrições destas opções dizem respeito aos ficheiros. Só os ficheiros podem ser pesquisados. Nas pastas e sistemas de ficheiros definidos pelo utilizador, pode especificar o atributo de pesquisa a ser atribuído a ficheiros criados nessa pasta ou no sistema de ficheiros definido pelo utilizador.

- Sim

O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.

- Não

O objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa.

**Nota:** Se a opção 'Pesquisar no próximo acesso após restauro do objecto' estiver seleccionada nos valores de sistema, quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.

- Só quando o objecto foi alterado

O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado desde a última pesquisa efectuada ao objecto. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado.

Se o atributo "Utilizar 'só quando os objectos foram alterados' para o valor de sistema pesquisa de controlo" **não** for especificado, este atributo 'só objecto alterado' não será utilizado e o objecto será pesquisado depois de modificado e quando o software de pesquisa indicar uma actualização.

**Nota:** Neste separador de ficheiros, pode também determinar o estado de pesquisa de um objecto.

**Nota:** Se a opção 'Pesquisar no próximo acesso após restauro do objecto' estiver seleccionada nos valores de sistema, quando um objecto com este atributo é restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.

---

## Informações relacionadas com o sistema de ficheiros integrado

Os tópicos que se seguem contêm informações relacionadas com o sistema de ficheiros integrado:

- "Chamada de Procedimento Remoto Independente de Transporte"
- "Referências" na página 125
- "Relatórios de Experiências" na página 126

## Chamada de Procedimento Remoto Independente de Transporte

Desenvolvida pela Sun Microsystems, a Chamada de Procedimento Remoto (RPC) separa e distribui facilmente aplicações cliente a partir de um mecanismo servidor. Esta chamada inclui uma norma para a representação de dados, denominada eXternal Data Representation, ou XDR, de modo a permitir a mais de um tipo de máquina aceder a dados transmitidos. A RPC Independente de Transporte (TI-RPC) é a última versão da RPC. Fornece um método para separar o protocolo subjacente que é utilizado ao nível da rede, possibilitando uma transição mais uniforme de um protocolo para outro. Os únicos protocolos que presentemente estão disponíveis no servidor iSeries são TCP e UDP.

O desenvolvimento de aplicações distribuídas numa rede é uma tarefa fácil quando a RPC é utilizada. Os alvos principais são aplicações que se destinam mais à distribuição da interface do utilizador ou à obtenção de dados.

Para obter mais informações sobre RPCs Independentes de Transporte, consulte os seguintes tópicos:

- "Seleções de rede"
- "Conversão nome-para-endereço" na página 121
- "eXternal Data Representation (XDR)" na página 121
- "Autenticação" na página 123
- "RPC Independente do Transporte (TI-RPC)" na página 123

## Seleções de rede

As APIs que se seguem fornecem o meio de selecção do transporte, em que uma aplicação deverá ser executada.

Estas APIs requerem a presença do ficheiro \*STMF /etc/netconfig no sistema. Se o ficheiro netconfig não existir no directório /etc, o utilizador terá de o copiar do directório /QIBM/ProdData/OS400/RPC. O ficheiro netconfig está sempre no directório /QIBM/ProdData/OS400/RPC.

API	Descrição
endnetconfig()	Liberta o indicador para os registos armazenados no ficheiro netconfig
freenetconfig()	Liberta a estrutura do netconfig que é devolvida da chamada à função getnetconfig()
getnetconfig()	Devolve o indicador ao registo actual no ficheiro netconfig e incrementa o respectivo indicador para o registo seguinte
getnetconfigent()	Devolve o indicador à estrutura do netconfig que corresponde ao netid de input
setnetconfig()	Inicializa o indicador de registo para a primeira entrada do ficheiro netconfig. A função setnetconfig() tem de ser utilizada antes da primeira utilização da função getnetconfig(). A função setnetconfig() devolve um único parâmetro identificador (um indicador para os registos armazenados no ficheiro netconfig) a ser utilizado pela função getnetconfig().

## Conversão nome-para-endereço

As APIs que se seguem permitem que uma aplicação obtenha o endereço de um serviço ou sistema central especificado de uma forma independente do transporte.

API	Descrição
netdir_free()	Liberta estruturas atribuídas por APIs de conversão de nome para endereço
netdir_getbyaddr()	Define endereços para nomes de sistemas centrais e de serviços
netdir_getbyname()	Define o nome do sistema central e do serviço especificados no parâmetro de serviço para um conjunto de endereços coerentes com o transporte identificado na estrutura do netconfig
netdir_options()	Fornecer interfaces para capacidades específicas do transporte como, por exemplo, o endereço de difusão e funções de porta reservada de TCP e UDP
netdir_spperror()	Emite uma mensagem informativa que indica porque uma das APIs de conversão de nome para endereço falhou
taddr2uaddr()	Converte um endereço específico do transporte (local) num endereço independente do transporte (universal)
uaddr2taddr()	Converte um endereço independente do transporte (universal) num endereço específico do transporte (local) (estrutura netbuf)

## eXternal Data Representation (XDR)

As APIs que se seguem permitem que aplicações da Chamada de Procedimento Remoto (RPC) tratem estruturas de dados arbitrarias, independentemente das diferentes ordens de bytes do respectivo sistema central ou convenções de esquema de estruturas.

API	Descrição
xdr_array()	Um valor primitivo de filtro que faz a conversão entre matrizes de comprimento variável e as respectivas representações externas correspondentes. Esta função é chamada para codificar ou decodificar cada elemento da matriz
xdr_bool()	Um valor primitivo de filtro que faz a conversão entre valores Booleanos (números inteiros em C) e as respectivas representações externas. Ao codificar dados, este filtro produz valores iguais a 1 ou 0.

API	Descrição
xdr_bytes()	Um valor primitivo de filtro que faz a conversão entre matrizes de bytes contados e as respectivas representações externas. Esta função trata um subconjunto de matrizes genéricas, em que o tamanho do elemento da matriz é conhecido como 1 e a descrição externa de cada elemento é incorporada. O comprimento da sequência de bytes está explicitamente localizado num número inteiro sem sinal. A sequência de bytes não termina com um carácter nulo. As representações interna e externa dos bytes são iguais.
xdr_char()	Um valor primitivo de filtro que faz a conversão entre caracteres de linguagem C e a respectiva representação externa
xdr_double()	Um valor primitivo de filtro que faz a conversão entre números de precisão dupla de linguagem C e as respectivas representações externas
xdr_double_char()	Um valor primitivo de filtro que faz a conversão entre caracteres de duplo byte de linguagem C e a respectiva representação externa
xdr_enum()	Um valor primitivo de filtro que faz a conversão entre a enumeração de linguagem C (enum) e a respectiva representação externa
xdr_free()	Liberta de modo recursivo o objecto indicado pelo indicador transferido
xdr_float()	Um valor primitivo de filtro que faz a conversão entre números de vírgula flutuante de linguagem C (números únicos de vírgula flutuante normalizados) e as respectivas representações externas
xdr_int()	Um valor primitivo de filtro que faz a conversão entre números inteiros de linguagem C e a respectiva representação externa
xdr_long()	Um valor primitivo de filtro que faz a conversão entre números inteiros longos de linguagem C e as respectivas representações externas
xdr_netobj()	Um valor primitivo de filtro que faz a conversão entre dados opacos de comprimento variável e a respectiva representação externa
xdr_opaque()	Um valor primitivo de filtro que faz a conversão entre dados opacos de comprimento fixo e a respectiva representação externa
xdr_pointer()	Possibilita a procura de indicadores em estruturas e coloca em série indicadores nulos. Pode representar estruturas de dados recursivas como, por exemplo, árvores binárias ou listas ligadas.
xdr_reference()	Um valor primitivo de filtro que possibilita a procura de indicadores em estruturas. Este valor primitivo permite colocar em série, retirar de série e libertar quaisquer indicadores, numa estrutura, que sejam referidos por outra estrutura. A função xdr_reference() não associa nenhum significado especial a um indicador nulo durante a colocação em série e transferir o endereço de um indicador nulo poderá causar um erro de memória. Por este motivo, o programador tem de descrever os dados com uma união bilateral discriminada. Um lado é utilizado quando o indicador é válido; o outro lado, quando o indicador é nulo.
xdr_short()	Um valor primitivo de filtro que faz a conversão entre números inteiros curtos de linguagem C e a respectiva representação externa
xdr_string()	Um valor primitivo de filtro que faz a conversão entre cadeias de linguagem C e as respectivas representações externas
xdr_u_char()	Um valor primitivo de filtro que faz a conversão entre caracteres de linguagem C sem sinal e as respectivas representações externas
xdr_u_int()	Um valor primitivo de filtro que faz a conversão entre números inteiros sem sinal de linguagem C e as respectivas representações externas
xdr_u_long()	Um valor primitivo de filtro que faz a conversão entre números inteiros longos sem sinal de linguagem C e as respectivas representações externas
xdr_u_short()	Um valor primitivo de filtro que faz a conversão entre números inteiros curtos sem sinal de linguagem C e as respectivas representações externas

API	Descrição
xdr_union()	Um valor primitivo de filtro que faz a conversão entre uniões de C discriminadas e as respectivas representações externas correspondentes
xdr_vector()	Um valor primitivo de filtro que faz a conversão entre matrizes de comprimento fixo e as respectivas representações externas correspondentes
xdr_void()	Não tem parâmetros. É transferido para outras funções de RPC que requerem um parâmetro, mas não transmite dados
xdr_wrapstring()	Um valor primitivo que chama a API xdr_string(xdr, sp, maxuint), onde maxuint é o valor máximo de um número inteiro sem sinal. xdr_wrapstring() é útil porque o pacote de RPC transfere um máximo de duas funções de XDR como parâmetros e a função xdr_string() requer três.

## Autenticação

As APIs que se seguem fornecem autenticação às aplicações da Chamada de Procedimento Remoto Independente do Transporte (TI-RPC).

API	Descrição
auth_destroy()	Destrói a estrutura das informações de autenticação que é indicada pelo parâmetro de autenticação
authnone_create()	Cria e devolve um parâmetro identificador de autenticação de RPC assumido que transfere informações de autenticação nulas em cada chamada de procedimento remoto.
authsys_create()	Cria e devolve um parâmetro identificador de autenticação de RPC que contém informações de autenticação

## RPC Independente do Transporte (TI-RPC)

As APIs que se seguem fornecem um ambiente de programação de aplicações distribuído isolando a aplicação de qualquer função de transporte específica. Esta função facilita a utilização de transportes.

Para obter mais informações sobre RPC Independente do Transporte (TI-RPC), consulte os seguintes tópicos:

- “APIs simplificadas TI-RPC”
- “APIs de nível superior TI-RPC”
- “APIs de nível intermédio TI-RPC” na página 124
- “APIs de nível de perito TI-RPC” na página 124
- “Outras APIs TI-RPC” na página 124

### APIs simplificadas TI-RPC:

As APIs simplificadas que se seguem especificam o tipo de transporte a utilizar. As aplicações que utilizam este nível não têm de criar explicitamente parâmetros identificadores.

API	Descrição
rpc_call()	Chama um procedimento remoto no sistema especificado
rpc_reg()	Regista um procedimento com o pacote de serviços de RPC

**APIs de nível superior TI-RPC:** As APIs que se seguem permitem que a aplicação especifique o tipo de transporte.

API	Descrição
clnt_call()	Chama um procedimento remoto associado ao cliente
clnt_control()	Altera as informações sobre um objecto cliente
clnt_create()	Cria um parâmetro identificador de cliente genérico
clnt_destroy()	Destrói o parâmetro identificador de RPC do cliente
svc_create()	Cria um parâmetro identificador de servidor
svc_destroy()	Destrói um parâmetro identificador de transporte de serviço de RPC

### APIs de nível intermédio TI-RPC:

As APIs que se seguem são semelhantes às APIs de nível superior, mas as aplicações do utilizador seleccionam as informações específicas do transporte, utilizando APIs de selecção da rede:

API	Descrição
clnt_tp_create()	Cria um parâmetro identificador de cliente
svc_tp_create()	Cria um parâmetro identificador de servidor

**APIs de nível de perito TI-RPC:** As APIs que se seguem permitem que a aplicação seleccione o transporte a utilizar. Também fornecem um nível de controlo melhorado sobre os detalhes dos parâmetros identificadores CLIENT e SVCXPRT. Estas APIs são semelhantes às APIs de nível intermédio com um controlo adicional fornecido, utilizando as APIs de conversão de nome para endereço.

É fornecido um controlo adicional, utilizando as APIs de conversão de nome para endereço.

API	Descrição
clnt_tli_create()	Cria um parâmetro identificador de cliente
rpcb_getaddr()	Localiza o endereço universal de um serviço
rpcb_set()	Regista o endereço do servidor com RPCbind
rpcb_unset()	É utilizada pelos servidores para anular do registo os respectivos endereços
svc_reg()	Associa o programa e a versão ao envio
svc_tli_create()	Cria um parâmetro identificador de servidor
svc_unreg()	Elimina um conjunto de associações por svc_reg()

**Outras APIs TI-RPC:** Estas APIs permitem que várias aplicações funcionem em coordenação com as APIs simplificadas, de nível superior, de nível intermédio e de nível de perito.

API	Descrição
clnt_freeres()	Liberta os dados atribuídos pela RPC ou pelo sistema XDR
clnt_geterr()	Obtém a estrutura de erro a partir do parâmetro identificador do cliente
svc_freeargs()	Liberta os dados atribuídos pela RPC ou pelo sistema XDR
svc_getargs()	Descodifica argumentos de um pedido de RPC
svc_getrpcaller()	Obtém o endereço de rede do programa de chamada
svc_run()	Aguarda a entrada de pedidos de RPC
svc_sendreply()	Envia os resultados de uma chamada de procedimento remoto para um cliente remoto.
svcerr_decode()	Envia informações para o cliente sobre o erro de descodificação

API	Descrição
svcerr_noproc()	Envia informações para o cliente sobre o erro do número de procedimentos
svcerr_systemerr()	Envia informações para o cliente sobre o erro do sistema

## Referências

Esta bibliografia lista as informações do servidor iSeries que contêm informações de segundo plano ou mais detalhes sobre as informações abordadas neste manual.

- O tópico Control language da categoria **Programming** do iSeries Information Center fornece uma descrição da linguagem de controlo (CL) do servidor iSeries e dos respectivos comandos. Cada descrição de comandos inclui um diagrama de sintaxe, parâmetros, valores assumidos, palavras-chave e um exemplo.
- O tópico Globalization do iSeries Information Center explica os conceitos de NLS (National Language Support), como um conjunto de caracteres e página de códigos e fornece informações necessárias para avaliar, planificar e utilizar o NLS do servidor iSeries e as capacidades multilingues.
- O tópico APIs da categoria **Programming** do iSeries Information Center fornece uma descrição de cada API do OS/400, incluindo as APIs do sistema de ficheiros integrado, bem como os pontos ou programas de saída do sistema de ficheiros integrado.
- O tópico Journal management da categoria **Systems management** do iSeries Information Center fornece informações sobre como configurar, gerir e resolver problemas relacionados com a protecção de caminhos de acesso geridos pelo sistema (SMAPP), diários locais e diários remotos num servidor iSeries.
- O tópico Commitment control da categoria **Database** do iSeries Information Center explica como definir e processar um grupo de alterações a recursos como, por exemplo, ficheiros de base de dados ou ficheiros do sistema de ficheiros integrado, como uma unidade de trabalho lógica.
- OS/400 Network File System Support  Este manual descreve o Sistema de Ficheiros da Rede através de uma série de aplicações práticas. Inclui também informações sobre como exportar, instalar, bloquear ficheiros e considerações sobre segurança. A partir deste manual, pode aprender a utilizar o NFS para criar e desenvolver um espaço de nome de rede protegida.
- Optical Support  Este manual constitui uma fonte de informações e de consulta para o utilizador sobre o IBM Optical Support no OS/400. As informações deste manual podem ajudar o utilizador a compreender os conceitos do servidor de dados da biblioteca óptica, a planear uma biblioteca óptica, a administrar e a funcionar com um servidor de dados da biblioteca óptica e a resolver problemas no servidor de dados ópticos.
- WebSphere Development Studio: ILE C/C++ Programmers Guide  Este manual fornece informações necessárias para conceber, editar, compilar, executar e depurar programas do ILE C/400 no servidor iSeries.
- WebSphere Development Studio: C/C++ Language Reference  Este manual fornece informações sobre a estrutura dos programas do ILE C/400 e contém detalhes sobre as funções de biblioteca, além de conter ficheiros (de cabeçalho).
- Security — Reference  Este manual fornece informações técnicas detalhadas sobre a segurança do OS/400, incluindo os valores de sistema relacionados com a segurança que afectam o processamento relacionado com a pesquisa do sistema de ficheiros integrado.
- APPC Programming  Este manual descreve o suporte de comunicações avançadas programa a programa (APPC) para o servidor iSeries. Constitui uma orientação para o desenvolvimento de programas de aplicação que utilizam APPC e para a definição do ambiente de comunicações para APPC.

- Backup and Recovery  Este manual fornece informações gerais sobre as opções de recuperação e disponibilidade relativas ao servidor iSeries IBM.

## **Relatórios de Experiências**

Os relatórios de experiências são escritos por programadores da IBM, que documentam as suas experiências na resolução de problemas, implementando cenários e soluções reais. Utilize-os para seguir as experiências dos programadores da IBM na implementação específica de uma solução iSeries, estão incluídas instruções passo a passo e sugestões.

O relatório de experiência que se segue está relacionado com ficheiros e sistemas de ficheiros.

Fazer uma cópia de segurança do Sistema de Ficheiros Integrado

---

## Apêndice. Informações especiais

Estas informações foram desenvolvidas para produtos e serviços disponibilizados nos E.U.A.

Os produtos, serviços ou componentes descritos neste documento poderão não ser disponibilizados pela IBM noutros países. Consulte o representante da IBM para obter informações sobre os produtos e serviços actualmente disponíveis na sua área. Quaisquer referências, nesta publicação, a produtos, programas ou serviços da IBM, não significam que apenas esses produtos, programas ou serviços da IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja qualquer direito de propriedade intelectual da IBM. No entanto, é da inteira responsabilidade do utilizador avaliar e verificar o funcionamento de qualquer produto, programa ou serviço de terceiros.

Nesta publicação, podem ser feitas referências a patentes ou a pedidos de patente pendentes da IBM. O facto de este documento lhe ser fornecido não lhe confere quaisquer direitos sobre essas patentes. Todos os pedidos de informação sobre licenças deverão ser endereçados a:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594-1785  
U.S.A.

Para endereçar os seus pedidos de informação sobre licenças relacionados com informações de conjunto de caracteres de duplo byte (DBCS - Double Byte Character Set), contacte o Departamento de Propriedade Intelectual da IBM no seu país ou envie-os, por escrito, para:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**O parágrafo seguinte não se aplica ao Reino Unido nem a qualquer outro país onde estas cláusulas sejam incompatíveis com a lei local:** A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "TAL COMO ESTÁ", SEM GARANTIA DE QUALQUER ESPÉCIE, QUER EXPLÍCITA QUER IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE NÃO INFRACÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO FIM. Alguns Estados não permitem a exclusão de garantias, quer explícitas quer implícitas, em determinadas transacções; esta declaração pode, portanto, não se aplicar ao seu caso.

É possível que estas informações contenham imprecisões técnicas ou erros de tipografia. A IBM permite-se fazer alterações periódicas às informações aqui contidas; essas alterações serão incluídas nas posteriores edições desta publicação. A IBM pode introduzir melhorias e/ou alterações ao(s) produto(s) e/ou programa(s) descrito(s) nesta publicação em qualquer altura, sem aviso prévio.

Quaisquer referências, nesta publicação, a sítios da Web de terceiros são fornecidas apenas para conveniência e não deverão nunca servir como aprovação desses sítios da Web. Os materiais existentes nesses sítios da Web não fazem parte dos materiais destinados a este produto da IBM e a utilização desses sítios da Web será da exclusiva responsabilidade do utilizador.

A IBM pode utilizar ou distribuir quaisquer informações que lhe sejam fornecidas pelo utilizador, de qualquer forma que julgue apropriada, sem incorrer em qualquer obrigação para com o autor dessas informações.

Os possuidores de licenças deste programa que pretendam obter informações sobre o mesmo com o objectivo de permitir: (i) a troca de informações entre programas criados de forma independente e outros programas (incluindo este) e (ii) a utilização recíproca das informações que tenham sido trocadas, deverão contactar:

IBM  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Tais informações poderão ser disponibilizadas, sujeitas a termos e condições apropriados, incluindo nalguns casos, ao pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para o programa são fornecidos pela IBM de acordo com os termos do IBM Customer Agreement, do IBM International Program License Agreement ou qualquer acordo equivalente entre ambas as partes.

Quaisquer dados de desempenho aqui contidos foram obtidos num ambiente controlado. Assim sendo, os resultados obtidos noutros ambientes operativos podem variar significativamente. Algumas medições podem ter sido efectuadas em sistemas de nível do desenvolvimento, pelo que não existem garantias de que estas medições sejam iguais nos sistemas normalmente disponíveis. Para além disso, algumas medições podem ter sido calculadas por extrapolação. Os resultados reais podem variar. Os utilizadores deste documento devem verificar os dados aplicáveis ao seu ambiente específico.

A informação relativa a produtos de terceiros foi obtida junto dos fornecedores desses produtos, dos seus comunicados ou de outras fontes de divulgação ao público. A IBM não testou esses produtos e não pode confirmar a exactidão do desempenho, da compatibilidade ou de quaisquer outras afirmações relacionadas com produtos produzidos por terceiros. Todas as questões sobre as capacidades dos produtos de terceiros deverão ser endereçadas aos fornecedores desses produtos.

Todas as declarações relativas às directivas ou intenções futuras da IBM estão sujeitas a alterações ou descontinuação sem aviso prévio, representando apenas metas e objectivos.

Todos os preços mostrados são os actuais preços de retalho sugeridos pela IBM e estão sujeitos a alterações sem aviso prévio. Os preços dos concessionários podem variar.

Estas informações destinam-se apenas a planeamento. As informações estão sujeitas a alterações antes de os produtos descritos ficarem disponíveis.

Estas informações contêm exemplos de dados e relatórios utilizados em operações comerciais diárias. Para ilustrá-los o melhor possível, os exemplos incluem nomes de indivíduos, firmas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e moradas reais é mera coincidência.

#### LICENÇA DE DIREITOS DE AUTOR:

Esta publicação contém programas de aplicação exemplo em linguagem-fonte, que ilustram técnicas de programação em diversas plataformas operativas. Pode copiar, modificar e distribuir estes programas exemplo de qualquer forma, sem encargos para com a IBM, com a finalidade de desenvolver, utilizar, comercializar ou distribuir programas de aplicação conformes à interface de programação de aplicações e destinados à plataforma operativa para a qual os programas exemplo são escritos. Estes exemplos não foram testados exaustivamente sob todas as condições. Deste modo, a IBM não garante nem se responsabiliza pela fiabilidade, assistência ou funcionamento destes programas. Pode copiar, modificar e distribuir estes programas exemplos de qualquer forma, sem encargos para com a IBM, com a finalidade de desenvolver, utilizar, comercializar ou distribuir programas de aplicação conformes às interfaces de programação de aplicações da IBM.

Cada cópia, qualquer parte destes programas exemplo ou qualquer trabalho derivado, deve incluir uma notificação de direitos de autor, conforme é mostrado a seguir:

© IBM Corp., 2004. Partes deste código derivam da IBM Corp. Programas exemplo. © Copyright IBM Corp. 2004. Todos os direitos reservados.

Se estiver a consultar estas informações em documentos electrónicos, é possível que as fotografias e as ilustrações a cores não estejam visíveis.

---

## **Informações sobre interfaces de programação**

Esta publicação documenta informações sobre Interfaces de Programação que permitem ao cliente escrever programas para obter os serviços do Sistema de Ficheiros Integrado.

---

## **Marcas comerciais**

Os termos seguintes são marcas comerciais da International Business Machines Corporation nos Estados Unidos e/ou noutros países:

Application System/400  
AS/400  
e (logótipo)  
IBM  
iSeries  
Operating System/400  
OS/400  
400  
C/400  
DB2  
Integrated Language Environment  
OfficeVision  
OS/2  
WebSphere  
xSeries

Lotus, Freelance e WordPro são marcas comerciais da International Business Machines Corporation e da Lotus Development Corporation nos Estados Unidos e/ou noutros países.

Microsoft, Windows, Windows NT e o logótipo do Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou noutros países.

UNIX é uma marca comercial registada do The Open Group nos Estados Unidos e noutros países.

Outros nomes de empresas, produtos e serviços podem ser marcas comerciais ou marcas de serviços de terceiros.

---

## **Termos e condições para descarregamento e impressão de publicações**

As permissões de utilização das informações seleccionadas para descarregamento são concedidas sujeitas aos seguintes termos e condições e à respectiva indicação de aceitação por parte do utilizador.

**Utilização pessoal:** Pode reproduzir estas publicações para uso pessoal e não comercial, desde que mantenha todas as informações de propriedade. Não pode realizar, distribuir ou apresentar qualquer trabalho derivado destas publicações, nem qualquer parte das mesmas, sem o expreso consentimento da IBM.

**Utilização comercial:** Pode reproduzir, distribuir e apresentar estas publicações exclusivamente no âmbito da sua empresa, desde que mantenha todas as informações de propriedade. Não pode realizar qualquer trabalho derivado destas publicações, nem reproduzir, distribuir ou apresentar estas publicações, ou qualquer parte das mesmas, fora das instalações da empresa, sem o expreso consentimento da IBM.

À excepção das concessões expressas nesta permissão, não são concedidos outros direitos, permissões ou licenças, quer explícitos, quer implícitos, sobre as publicações ou quaisquer informações, dados, software ou outra propriedade intelectual contidos nesta publicação.

A IBM reserva-se o direito de retirar as permissões concedidas nesta publicação sempre que considerar que a utilização das Publicações pode ser prejudicial aos seus interesses ou, tal como determinado pela IBM, sempre que as instruções acima referidas não estejam a ser devidamente cumpridas.

Não pode descarregar, exportar ou reexportar estas informações, excepto quando em total conformidade com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação em vigor nos Estados Unidos. A IBM NÃO FORNECE QUAISQUER GARANTIAS RELATIVAMENTE AO CONTEÚDO DESTAS PUBLICAÇÕES. AS PUBLICAÇÕES SÃO FORNECIDAS "TAL COMO ESTÃO" E SEM GARANTIAS DE QUALQUER ESPÉCIE, QUER EXPLÍCITAS, QUER IMPLÍCITAS, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO FIM.

Todo o material está protegido por direitos de autor da IBM Corporation.

Ao descarregar ou imprimir publicações a partir deste sítio da Web, o utilizador indica que concorda com estes termos e condições.



**IBM**