



iSeries

# Sistema de Ficheiros Integrado - Introdução

*Versão 5*







@server

iSeries

Sistema de Ficheiros Integrado - Introdução

*Versão 5*



# Índice

<b>Acerca da Introdução ao Sistema de Ficheiros Integrado</b> . . . . .	vii
Quem deve ler o manual Introdução ao Sistema de Ficheiros Integrado . . . . .	vii
Informações de renúncia de código . . . . .	vii
<b>Capítulo 1. Introdução ao sistema de ficheiros integrado</b> . . . . .	1
O que é o sistema de ficheiros integrado . . . . .	1
Porquê utilizar o sistema de ficheiros integrado? . . . . .	1
<b>Capítulo 2. Conceitos do sistema de ficheiros integrado</b> . . . . .	3
Ficheiro de dados contínuos . . . . .	3
Ficheiros de dados contínuos *TYPE1 e *TYPE2 . . . . .	4
Sistemas de ficheiros no sistema de ficheiros integrado . . . . .	4
Directório . . . . .	6
Directório actual e directório inicial . . . . .	8
Directórios *TYPE2 . . . . .	9
Utilizar directórios *TYPE2 no OS/400 V5R1 . . . . .	11
Converter num directório *TYPE2 . . . . .	11
Indisponibilidade do sistema de ficheiros "Raiz", QOpenSys ou UDFS . . . . .	11
Requisitos de memória auxiliar . . . . .	12
Considerações sobre ligações simbólicas . . . . .	12
Conjuntos de memória auxiliar (ASPs) independentes . . . . .	12
Considerações sobre guardar/restaurar . . . . .	13
Preparar conversão em *TYPE2 . . . . .	13
Processamento da conversão . . . . .	14
Exemplo: Converter todos os sistemas de ficheiros (pequeno número de objectos) . . . . .	16
Exemplo: Converter todos os sistemas de ficheiros (grande número de objectos) . . . . .	16
Exemplo: Converter apenas certos ASPs . . . . .	17
Nome de caminho . . . . .	17
Ligação . . . . .	18
Ligação permanente . . . . .	19
Ligação simbólica . . . . .	20
Comparação: Ligação permanente e ligação simbólica . . . . .	21
Atributos expandidos . . . . .	22
Continuidade de nomes . . . . .	23
<b>Capítulo 3. Aceder ao sistema de ficheiros integrado utilizando a interface de sistema tradicional</b> . . . . .	25
Executar operações utilizando menus e ecrãs do iSeries . . . . .	25
Executar operações utilizando comandos de CL . . . . .	26
Regras de nomes de caminho para comandos de CL e ecrãs . . . . .	29
Executar operações utilizando um PC . . . . .	31
Transferir ficheiros utilizando FTP . . . . .	32
Trabalhar com ficheiros utilizando o iSeries NetServer . . . . .	33
Mover objectos para outro sistema de ficheiros . . . . .	34
Considerações sobre a movimentação de objectos para outro sistema de ficheiros . . . . .	34
Directórios fornecidos pelo sistema de ficheiros integrado . . . . .	35
<b>Capítulo 4. Aceder ao sistema de ficheiros integrado com o iSeries Navigator</b> . . . . .	37
Dar entrada de um ficheiro . . . . .	37
Dar saída de um ficheiro . . . . .	37
Configurar permissões para um ficheiro ou arquivador . . . . .	38
Configurar conversão de texto do ficheiro . . . . .	38
Enviar um ficheiro ou arquivador para outro sistema . . . . .	39

Alterar opções para a definição do pacote . . . . .	39
Marcar uma data e uma hora para enviar o ficheiro ou arquivador . . . . .	40
Criar um arquivador . . . . .	40
Remover um arquivador . . . . .	40
Criar uma partilha de ficheiros. . . . .	40
Alterar uma partilha de ficheiros . . . . .	41
Criar um novo sistema de ficheiros definido pelo utilizador . . . . .	41
Instalar um sistema de ficheiros definido pelo utilizador . . . . .	41
Desinstalar um sistema de ficheiros definido pelo utilizador . . . . .	42
Iniciar registo em diário . . . . .	42
Terminar registo em diário . . . . .	43
<b>Capítulo 5. Programar suporte para o sistema de ficheiros integrado.</b> . . . . .	<b>45</b>
Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados. . . . .	45
Copiar dados utilizando comandos de CL . . . . .	46
Copiar dados utilizando APIs . . . . .	47
Copiar dados utilizando a função de transferência de dados. . . . .	47
Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda. . . . .	49
Executar operações utilizando APIs . . . . .	50
Funções de ILE C/400 . . . . .	55
Suporte de ficheiros grandes para APIs . . . . .	56
Regras de nomes de caminho para APIs . . . . .	56
Descritor de ficheiros . . . . .	57
Segurança . . . . .	58
Suporte de sockets . . . . .	58
Nomenclatura e suporte internacional . . . . .	59
Conversão de dados . . . . .	59
<b>Capítulo 6. Sistemas de ficheiros no sistema de ficheiros integrado</b> . . . . .	<b>61</b>
Comparação entre sistemas de ficheiros . . . . .	61
Sistema de ficheiros “raiz” (/) . . . . .	64
Utilizar o sistema de ficheiros “raiz” (/). . . . .	65
Sistema de ficheiros de sistemas abertos (QOpenSys). . . . .	66
Utilizar o QOpenSys . . . . .	66
Sistema de ficheiros definido pelo utilizador (UDFS). . . . .	67
Conceitos de UDFS . . . . .	68
Utilizar o UDFS através da interface do sistema de ficheiros integrado . . . . .	68
Sistema de ficheiros de biblioteca (QSYS.LIB). . . . .	71
Utilizar o QSYS.LIB através da interface do sistema de ficheiros integrado . . . . .	72
QSYS.LIB de ASP Independente. . . . .	74
Utilizar QSYS.LIB de ASP Independente através da interface do sistema de ficheiros integrado . . . . .	74
Sistema de Ficheiros de Serviços da Biblioteca de Documentos (QDLS) . . . . .	77
Utilizar o QDLS através da interface do sistema de ficheiros integrado . . . . .	77
Sistema de Ficheiros Óptico (QOPT) . . . . .	79
Utilizar o QOPT através da interface do sistema de ficheiros integrado . . . . .	79
Sistema de ficheiros NetWare (QNetWare) . . . . .	81
Montar sistemas de ficheiros de NetWare . . . . .	81
Estrutura de directórios QNetWare . . . . .	82
Utilizar QNetWare através da interface do sistema de ficheiros integrado . . . . .	82
Sistema de Ficheiros do Windows NT Server (QNTC) . . . . .	84
Utilizar o QNTC através da interface do sistema de ficheiros integrado . . . . .	84
Sistema de Ficheiros do Servidor de Ficheiros do OS/400 (QFileSvr.400). . . . .	86
Utilizar o QFileSvr.400 através da interface do sistema de ficheiros integrado . . . . .	86
Sistema de Ficheiros da Rede (NFS) . . . . .	90
Utilizar sistemas de ficheiros NFS através da interface do sistema de ficheiros integrado . . . . .	90

<b>Capítulo 7. Suporte de registo em diário para objectos do sistema de ficheiros integrado</b>	93
Gestão de registos em diário	93
Objectos que deve registar em diário	93
Objectos do sistema de ficheiros integrado registados em diário	94
Operações registadas em diário	95
Considerações especiais para entradas de diário	95
<b>Apêndice A. Chamada de Procedimento Remoto Independente de Transporte</b>	97
Seleccões de rede	97
Conversão nome-para-endereço	97
eXternal Data Representation (XDR)	98
Autenticação	99
RPC Independente do Transporte (TI-RPC)	100
APIs simplificadas TI-RPC	100
APIs de nível superior TI-RPC	100
APIs de nível intermédio TI-RPC	100
APIs de nível de perito TI-RPC	100
Outras APIs TI-RPC	101
<b>Apêndice B. Programa exemplo que utiliza funções C do sistema de ficheiros integrado</b>	103
<b>Apêndice C. Exemplo de código RPG do sistema de ficheiros integrado</b>	109
<b>Bibliografia</b>	111
<b>Índice Remissivo</b>	113





---

# Acerca da Introdução ao Sistema de Ficheiros Integrado

Este manual apresenta uma descrição geral do sistema de ficheiros integrado, que inclui:

- O que é o sistema de ficheiros integrado?
- Porque razão poderá querer utilizá-lo.
- Conceitos e terminologia do sistema de ficheiros integrado.
- As interfaces que pode utilizar para comunicar com o sistema de ficheiros integrado.
- As APIs e as técnicas que pode utilizar para criar programas que comuniquem com o sistema de ficheiros integrado.
- Características dos sistemas de ficheiros individuais.

---

## Quem deve ler o manual Introdução ao Sistema de Ficheiros Integrado

Este manual destina-se a utilizadores, programadores e gestores do servidor iSeries que pretendam compreender o sistema de ficheiros integrado e a forma como pode ser utilizado.

---

## Informações de renúncia de código

Este documento contém exemplos de programação.

A IBM concede ao Adquirente uma licença de copyright 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.

Todo o código exemplo é fornecido pela IBM apenas para fins ilustrativos. Estes exemplos não foram testados exaustivamente sob todas as condições. Desta forma, a IBM não pode garantir o 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.



---

# Capítulo 1. Introdução ao 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

O **sistema de ficheiros integrado** é uma parte do OS/400 que suporta input/output de dados contínuos 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.

As funções-chave do sistema de ficheiros integrado são:

- Suporte para armazenar informações em ficheiros de dados 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, no Integrated xSeries Server for iSeries ou num servidor de Windows NT remoto. Os ficheiros de dados contínuos também podem ser armazenados remotamente num servidor de rede local (LAN), num servidor Novell NetWare e noutro servidor iSeries remoto ou num servidor de Sistema de Ficheiros de Rede.

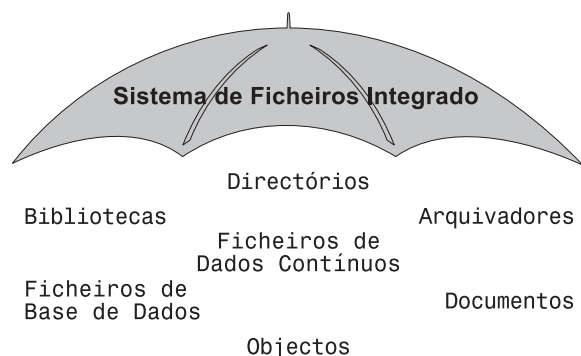


Figura 1. Estrutura de todas as informações armazenadas no servidor iSeries

---

## 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 multimédia.

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 Disk Operating System (DOS) e Windows.
- 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.

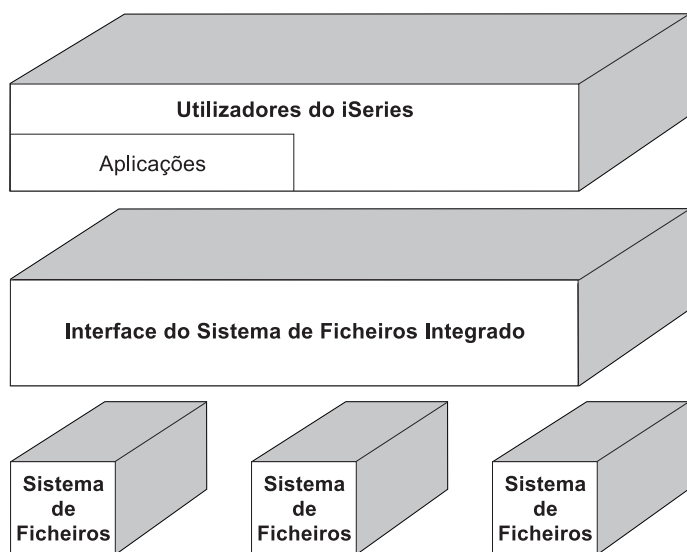


Figura 2. Interface comum para sistemas de ficheiros separados

- 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 trabalhar com ficheiros de dados contínuos do servidor iSeries e outros objectos tal como trabalham com os ficheiros armazenados nos seus 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.

---

## Capítulo 2. Conceitos do sistema de ficheiros integrado

---

### 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 nos arquivadores do sistema são ficheiros de dados contínuos. Outros exemplos de ficheiros de dados contínuos são os ficheiros de PC e os ficheiros nos 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 3. 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 na 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.

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 45.
- "Ficheiros de dados contínuos \*TYPE1 e \*TYPE2" na página 4.

## Ficheiros de dados contínuos \*TYPE1 e \*TYPE2

Um ficheiro tem uma de duas opções de formato: ficheiro de dados contínuos \*TYPE1 ou \*TYPE2.

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. É guardado mais rapidamente que um ficheiro de dados contínuos \*TYPE2 em edições anteriores à versão 4, edição 4, do OS/400. Tem um tamanho mínimo de 4096 bytes.

Um ficheiro de dados contínuos \*TYPE2 tem um acesso a ficheiros de alto rendimento e era novo na versão 4, edição 4, do OS/400. É guardado mais lentamente que um ficheiro de dados contínuos \*TYPE1 em edições anteriores à versão 4, edição 4, do OS/400. Tem um tamanho de objecto mínimo de 4096 bytes. Todos os ficheiros criados em sistemas com a V4R4 e versões mais recentes são ficheiros de dados contínuos \*TYPE2.

Embora os ficheiros de dados contínuos \*TYPE2 só funcionem em sistemas com a V4R4 e versões mais recentes, pode guardar um ficheiro de dados contínuos \*TYPE2 para restauro num sistema com uma versão anterior à V4R4 instalada. No entanto, este processo pode ser demorado.

---

## Sistemas de ficheiros no sistema de ficheiros integrado

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 a informação em memória. Estas estruturas e regras podem ser diferentes de um sistema de ficheiros para outro. De facto, da 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 de OS/400 para aceder a documentos (que são, na realidade, ficheiros de dados contínuos) através da estrutura de arquivadores pode ser considerado como um sistema de ficheiros separado.

O sistema de ficheiros integrado considera o suporte de bibliotecas e o suporte de arquivadores 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 61.

Os sistemas de ficheiros no sistema de ficheiros integrado são:

### "raiz" (/)

O sistema de ficheiros "**raiz**" (/). Este sistema de ficheiros tira o máximo partido do suporte de ficheiro 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 Disk Operating System (DOS) e do OS/2.

### QOpenSys

Sistema de ficheiros de sistemas abertos. Este sistema de ficheiros é compatível com as normas de sistema aberto com base em UNIX, tais 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 directórios que é fornecido pelo sistema de ficheiros integrado. Para além disso, suporta nomes de objectos sensíveis a maiúsculas e minúsculas.

**UDFS** Sistema de ficheiros definido pelo utilizador. Este sistema de ficheiros reside no conjunto de memória auxiliar (ASP) ou num conjunto de memória auxiliar (ASP) independente à sua escolha. É o utilizador que cria e gere este sistema de ficheiros.

**QSYS.LIB**

Sistema de ficheiros de biblioteca. Este sistema de ficheiros suporta a estrutura de bibliotecas do servidor. Este sistema de ficheiros fornece acesso a ficheiros de base de dados e a todos os outros tipos de objectos do Servidor de iSeries geridos pelo suporte de bibliotecas no ASP do sistema e nos ASPs base do utilizador.

**QSYS.LIB de ASP Independente**

O sistema de ficheiros QSYS.LIB de ASP Independente. Este sistema de ficheiros suporta a sua estrutura de biblioteca do servidor em quaisquer 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 Servidor de iSeries geridos pelo suporte de bibliotecas.

**QDLS** Sistema de ficheiros de serviços da biblioteca de documentos. Este sistema de ficheiros permite o acesso a documentos e arquivadores.

**QOPT**

Sistema de ficheiros ópticos. Este sistema de ficheiros permite o acesso a dados contínuos armazenados em suportes ópticos.

**QNetWare**

Sistema de ficheiros QNetWare. Este sistema de ficheiros permite o acesso a dados locais ou remotos e a objectos que estejam armazenados num servidor que esteja a utilizar o Novell NetWare 4.10 ou 4.11 ou a servidores de PC autónomos que estejam a utilizar o Novell NetWare 3.12, 4.10, 4.11 ou 5.0. Pode instalar de modo dinâmico sistemas de ficheiros NetWare sobre sistemas de ficheiros locais existentes.

**QNTC** Sistema de ficheiros do Windows NT Server. Este sistema de ficheiros permite o acesso aos dados e objectos que estão armazenados num servidor que esteja a utilizar o Windows NT 4.0 ou superior. Permite às aplicações do servidor iSeries utilizar os mesmos dados que os clientes do Windows NT. Isto inclui o acesso aos dados num servidor Windows NT Server que esteja em execução num servidor de PC integrado. Consulte o manual OS/400-AS/400 Integration with Windows NT Server, SC41-5439-01 (SC41-5439) para obter mais detalhes.

**QFileSvr.400**

Este sistema de ficheiros permite o acesso a outros sistemas de ficheiros que residam em servidores iSeries remotos.

**NFS** Sistema de Ficheiros da Rede. Este sistema de ficheiros permite-lhe ter acesso 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.

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

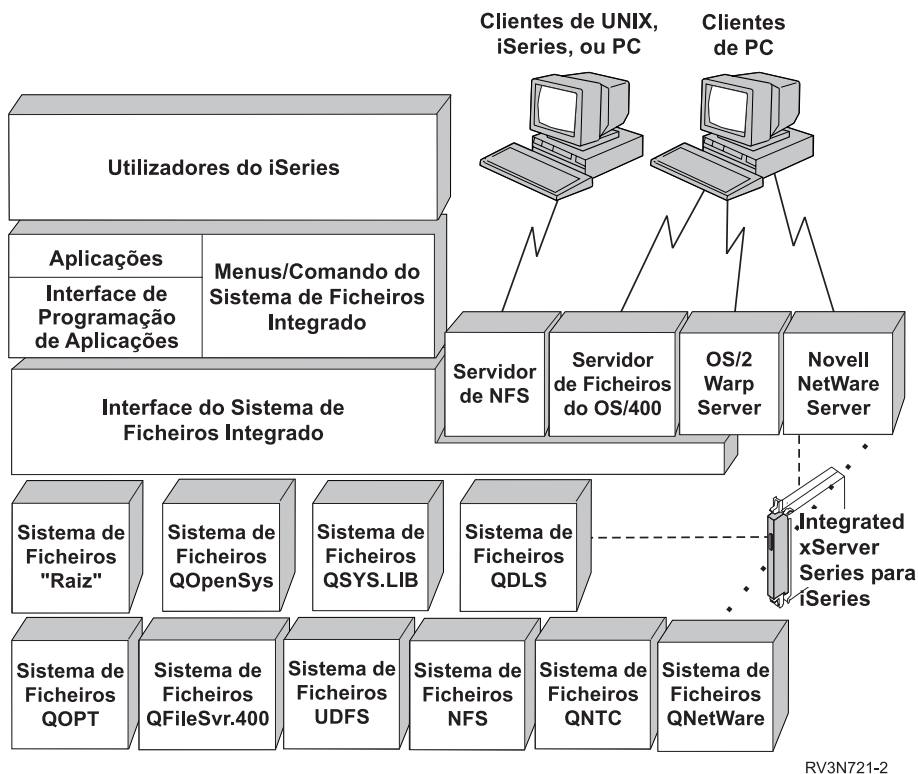




Figura 4. Sistemas de ficheiros, servidores de ficheiros e a interface do sistema de ficheiros integrado

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

- Optical Support 
- OS/400 Network File System Support 

## 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), os arquivadores são tratados como subdirectórios e os documentos em arquivadores 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 pode 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 e directório inicial” na página 8
- “Directórios fornecidos pelo sistema de ficheiros integrado” na página 35
- “Directórios \*TYPE2” na página 9

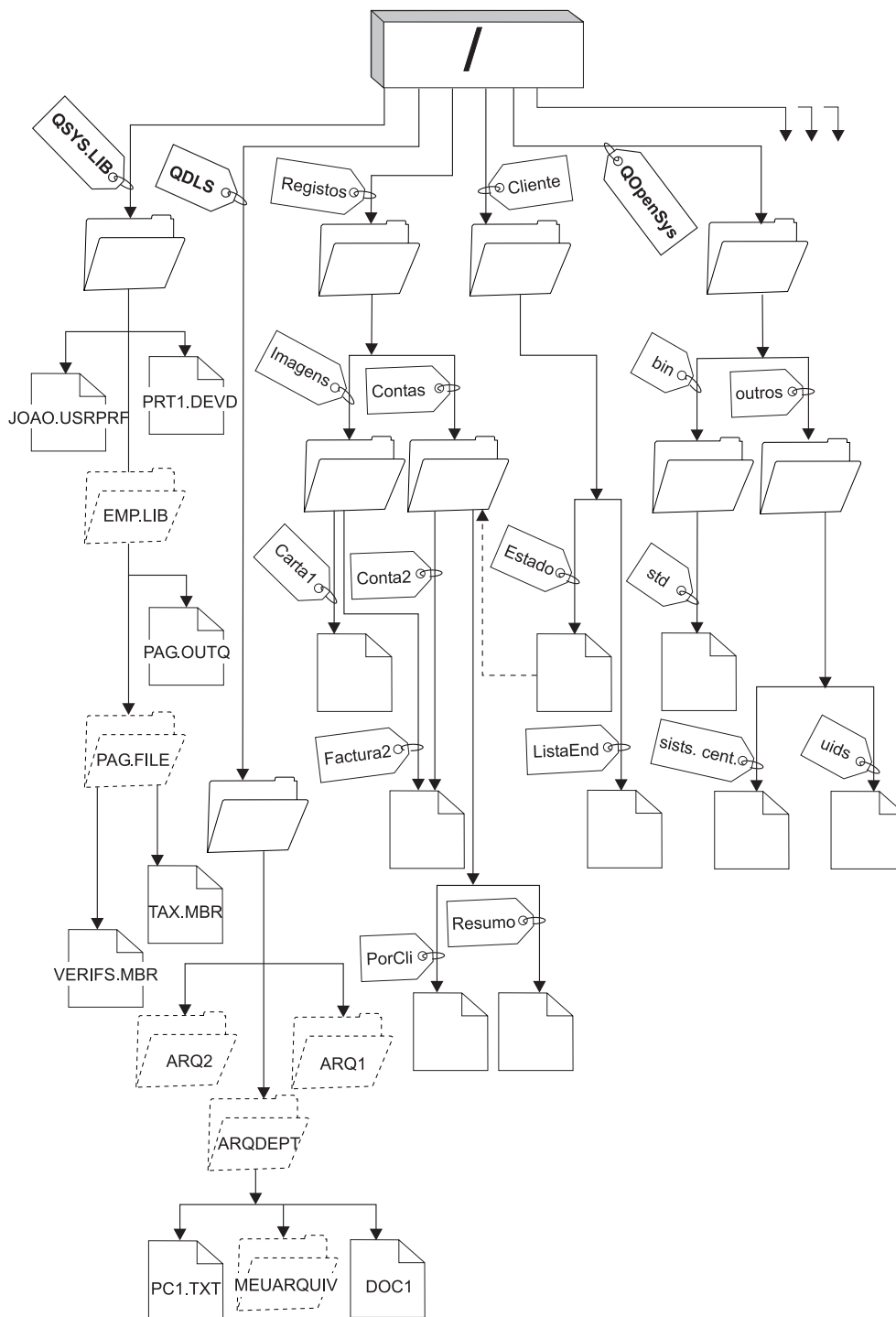


Figura 5. Os sistemas de ficheiros e objectos são ramos da árvore de directórios do sistema de ficheiros integrado

## Directório actual e directório inicial

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**.

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/JOA0. Se o directório /home/JOA0 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 (`CHGCDIR`), 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 \*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 1 milhão de subdirectórios.
3. Utilizando o iSeries Navigator, a lista de entradas é automaticamente ordenada em sequência binária quando abre um directório com o formato \*TYPE2.

Normalmente, os directórios \*TYPE2 que tenham 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 para o seu sistema:

- Um sistema de ficheiros definido pelo utilizador (UDFS) num conjunto de memória auxiliar (ASP) independente é convertido no formato \*TYPE2 na primeira vez que o ASP Independente for activado num sistema instalado com o OS/400 V5R2.
- Todos os outros sistemas de ficheiros suportados, excepto o UDFSs em ASPs independentes têm de ser convertidos em \*TYPE2 com a utilização do comando Converter Directório (CVTDIR).
- Os novos servidores iSeries que sejam pré-carregados com o OS/400 V5R2 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 num Servidor de iSeries tem directórios \*TYPE2. Não é necessária a conversão do directório "raiz" (/), do QOpenSys e dos UDFSs nos ASPs 1-32.

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, mas existem algumas diferenças do suporte de directório \*TYPE2 normal. Para obter mais informações, consulte Utilizar directórios \*TYPE2 no OS/400 V5R1.

Para obter mais informações sobre directórios \*TYPE2, consulte os seguintes tópicos:

- Converter num directório \*TYPE2
- Indisponibilidade de "Raiz", QOpenSys ou UDFS
- Requisitos de memória auxiliar
- Considerações sobre ligações simbólicas
- Conjuntos de memória auxiliar (ASPs) independentes
- Considerações sobre guardar/restaurar
- Preparar para a conversão \*TYPE2
- Processamento da conversão
- Exemplo: Converter todos os sistemas de ficheiros (pequeno número de objectos)
- Exemplo: Converter todos os sistemas de ficheiros (grande número de objectos)
- Exemplo: Converter apenas certos ASPs

## Utilizar directórios \*TYPE2 no OS/400 V5R1

Os sistemas de ficheiros "raiz" (/), QOpenSys e os sistemas de ficheiros definidos pelo utilizador (UDFS) do sistema de ficheiros integrado suportam o formato de directório \*TYPE2 no 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.


Se tiver a V5R1, poderá converter os seus directórios V5R1 para o formato de directório \*TYPE2. É recomendável converter no formato de directório \*TYPE2 **antes** de instalar uma nova edição do OS/400. Isto é necessário porque a conversão de directório pode ser automaticamente executada durante a instalação. O impacto de uma conversão automática durante a instalação corresponde a um aumento **significativo** do tempo necessário para a instalação.

**Nota:** Se actualizar para o OS/400 V5R1 ou V5R2, a conversão automática para o formato de directório \*TYPE2 **não** ocorre. Não é necessário converter os directórios antes destas instalações.

O suporte para os 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 I113161 para obter a documentação completa sobre os directórios \*TYPE2 na V5R1. Utilize um dos seguintes métodos para ter acesso ao APAR:

1. Transfira o APAR informativo para o seu Servidor de iSeries e consulte-o. Utilize os seguintes comandos:

```
SNDPTFORD PTFID((I113161))  
DSPPTFCVR LICPGM(INFOAS4) SELECT(I113161)
```

2. Vá para <http://www-912.ibm.com>  para visualizar o APAR informativo online. Selecione **Authorized Program Analysis Reports (APARs) → V5R1 APARs → APAR número I113161**.

## Converter num directório \*TYPE2

O comando CVTDIR executa a conversão de um directório \*TYPE1 num directório \*TYPE2. Adicionalmente, fornece informações sobre como converter os sistemas de ficheiros no formato de directório \*TYPE2. O CVTDIR executa o seguinte:

- Lista o formato de directório actual para os sistemas de ficheiros existentes que suportam directórios \*TYPE2.
- Estima o tempo que a conversão irá demorar.
- Estima os requisitos de memória auxiliar para a conversão.
- Converte os sistemas de ficheiros para o formato \*TYPE2. Todos os directórios existentes são convertidos em \*TYPE2 e todos os novos directórios criados após a conversão serão \*TYPE2.

Existem várias formas de converter os directórios num dos sistemas de ficheiros:

- Manualmente, utilizando o comando CVTDIR
- Automaticamente, na primeira vez que um ASP independente é activado num sistema que tenha o OS/400 V5R2 instalado
- Durante um IPL, se o sistema determinar que a conversão de um sistema de ficheiros estava em curso quando o sistema terminou anormalmente
- Durante a operação Regenerar Memória (RCLSTG SELECT(\*ALL)), caso sejam encontrados directórios \*TYPE1 perdidos que façam parte de um sistema de ficheiros que tenha sido convertido no formato \*TYPE2

## Indisponibilidade do sistema de ficheiros "Raiz", QOpenSys ou UDFS

A conversão dos sistemas de ficheiros "raiz" (/) ou QOpenSys tem de ser executada quando o sistema está num estado restrito. Ao converter UDFSs, o sistema não é obrigado a estar num estado restrito; no

entanto, os UDFSs nesse ASP não estarão disponíveis durante a conversão. O período de tempo necessário para executar uma conversão depende do tamanho do sistema de ficheiros. Deste modo, é necessário um planeamento para programar o melhor momento para executar a conversão. A opção \*ESTIMATE do comando CVTDIR estima o período de tempo necessário para converter o sistema de ficheiros especificado. O período de tempo estimado é o valor máximo estimado. Estima o período de tempo com base numa conversão executada num trabalho com um único módulo. A conversão real utiliza vários módulos e deverá demorar menos tempo do que o estimado. Normalmente, os sistemas de ficheiros com mais de 40.000 ligações podem ser convertidos em 30 a 50 por cento do tempo estimado. No entanto, o tempo real depende do hardware e da configuração do servidor.

Se o sistema costuma terminar anormalmente durante a execução do comando CVTDIR, no próximo IPL, a função de conversão será executada no trabalho SCPF. O trabalho SCPF não permite que estejam vários módulos activos. Assim, quando a conversão de um sistema de ficheiros tiver de ser concluída durante o IPL, ela será executada com a utilização de um único módulo. A função de conversão é executada quando é apresentado o SRC C900 2A85 durante o IPL e é apresentada a mensagem de estado CPIA089, indicando o progresso da conversão.

A função de conversão é executada durante RCLSTG caso existam directórios\*TYPE1 perdidos de um sistema de ficheiros já convertido em \*TYPE2. A função de conversão é executada no trabalho que emite o comando RCLSTG. Se forem encontrados directórios perdidos que necessitem de ser convertidos, a conversão será executada num único módulo devido a restrições do sistema.

## Requisitos de memória auxiliar

Devem ser considerados requisitos de memória auxiliar antes da conversão dos 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 no 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 um determinado número de directórios tenha uma versão \*TYPE1 e uma versão \*TYPE2 simultaneamente. Este número depende da configuração do Servidor de iSeries e da estrutura de directórios do sistema de ficheiros que estiver a ser convertido.

A opção \*ESTIMATE do comando CVTDIR fornece informações que indicam a estimativa da quantidade de memória auxiliar necessária durante a conversão.

## Considerações sobre ligações simbólicas

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 é passível de alteração. 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.

## Conjuntos de memória auxiliar (ASPs) independentes

Na primeira vez que um ASP independente é activado num sistema instalado com o OS/400 V5R2, os directórios são convertidos em \*TYPE2. Para fins de planeamento, é fornecida uma função de estimativa no OS/400 V5R1 que dá informações sobre o período de tempo que irá demorar uma conversão. Antes

de activar o ASP independente para o servidor com a V5R2, execute a seguinte API no seu sistema com a V5R1 quando o ASP independente (designado ASP\_NAME) for activado:

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

**Nota:** Recomenda-se que execute RCLSTG no ASP independente no sistema com a V5R1 antes de chamar esta função.

## Considerações sobre guardar/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.

## Preparar conversão em \*TYPE2

Existem vários comandos e parâmetros de CL que são recomendados antes de converter directórios para \*TYPE2:

- Regenerar Memória (RCLSTG)

Utilizando o comando RCLSTG SELECT(\*ALL) antes de converter qualquer sistema de ficheiros, limpa os directórios e assegura que estes estão em boas condições. Embora esta acção não elimine todos os problemas possíveis que possam ser encontrados durante uma conversão de directório, assegura que os directórios do sistema de ficheiros podem ser lidos.

Este comando só tem de ser executado uma vez antes de ser utilizada qualquer uma das opções do comando CVTDIR.

- Guardar Sistema (SAVSYS)

Deverá ser executada uma salvaguarda total do Servidor de iSeries após a execução de RCLSTG e antes da utilização da opção \*CONVERT do comando CVTDIR. Para efectuar uma cópia de segurança do seu sistema, utilize o menu Guardar do iSeries. Para ir para o menu Guardar, escreva G0 SAVE em qualquer linha de comandos e seleccione a opção 21. Para obter mais informações, consulte Cópia de

Segurança e Recuperação .

Este comando só tem de ser executado uma vez antes de ser utilizada qualquer uma das opções do comando CVTDIR.

- A opção \*ESTIMATE do comando CVTDIR

Utilize a opção \*ESTIMATE do comando CVTDIR para determinar o tempo necessário para converter os directórios.

Para além de fornecer estimativas sobre o tempo e a memória auxiliar, a opção \*ESTIMATE traz benefícios adicionais. Constrói certos objectos secundários associados aos directórios\*TYPE1, o que permite que a conversão seja executada mais rapidamente (uma vez que não necessita de os criar). Estes objectos secundários permanecem até ser utilizada a opção \*CONVERT para converter o sistema de ficheiros no formato \*TYPE2. A opção \*ESTIMATE também lê todos os directórios do sistema de ficheiros, verificando-os implicitamente. A opção \*ESTIMATE não garante encontrar todos os erros que possam ocorrer durante a conversão real, mas ajuda.

Após ter executado a opção \*ESTIMATE, verifique se existem erros no registo de trabalhos e execute quaisquer acções de recuperação recomendadas antes de converter o sistema de ficheiros. Não é necessário executar de novo a estimativa após a acção de recuperação recomendada, mas ela pode ser executada opcionalmente para se verificar se não são encontrados outros problemas.

- Considerações sobre memória auxiliar

Verifique a memória auxiliar disponível para o ASP que contém o sistema de ficheiros que está a ser convertido. A opção CVTDIR \*ESTIMATE apresenta a mensagem CPIA090, que indica quanta memória auxiliar está disponível para o ASP. Adicionalmente, apresenta a estimativa da quantidade de memória auxiliar necessária durante a conversão. A mensagem CPIA091 também é apresentada, indicando se o



tamanho total dos directórios \*TYPE2 no sistema de ficheiros após a conversão está estimado como sendo maior ou menor do que o dos directórios \*TYPE1 existentes. A memória disponível no ASP deverá ser a soma da memória auxiliar não utilizada (apresentada na mensagem CPIA090) e a diferença entre os tamanhos dos directórios \*TYPE1 e \*TYPE2 (apresentada na mensagem CPIA091). Como alternativa, poderá encontrar a memória auxiliar disponível utilizando o comando Iniciar Ferramentas de Serviço do Sistema (STRSST) e seleccionando a opção Trabalhar com Unidades de Discos.

**Nota:** Se só existir um APS definido no sistema, o comando Trabalhar com Estado do Sistema (WRKSYSSTS) é suficiente para mostrar informações sobre o conjunto de memória auxiliar disponível.

É recomendável executar uma limpeza geral do sistema antes de utilizar qualquer uma das opções do comando CVTDIR. Se existirem directórios ou ficheiros desnecessários, remova-os antes de utilizar quaisquer opções do comando CVTDIR. Essa operação libertará espaço na memória auxiliar, fornecerá uma estimativa mais precisa do espaço de memória auxiliar disponível e permitirá a conclusão da conversão em menos tempo porque existem menos objectos que necessitam de processamento.

- Considere alterar a acção de fila de mensagens de trabalho cheia para \*PRTWRAP para o trabalho que estiver a emitir o comando CVTDIR. Esta operação executa o seguinte:
  1. Impede que o trabalho termine anormalmente no caso de o registo de trabalhos ficar cheio
  2. Imprime quaisquer mensagens sobrepostas num ficheiro em Spool, caso ocorra um reinício cíclico do registo de trabalhos; deste modo, não se perdem mensagens importantes
- Nos sistemas com ASPs independentes: antes de activar o ASP independente num sistema a executar o OS/400 V5R2, utilize a função \*ESTIMATE da V5R1 em todos os ASPs independentes. Deste modo, fornecerá uma estimativa do tempo que irá demorar a primeira activação do ASP independente após a instalação. Consulte Conjuntos de memória auxiliar (ASPs) independentes para obter mais informações.

## Processamento da conversão

O comando CVTDIR converte os directórios \*TYPE1 em directórios \*TYPE2. Existem vários aspectos a considerar durante o processo de conversão:

- Converter "raiz" (/) ou QOpenSys
- Converter sistemas de ficheiros definidos pelo utilizador
- Criar perfis de utilizador
- Objectos com o nome mudado
- Considerações sobre perfis de utilizador

### Converter "raiz" ou QOpenSys

Quando converte os sistemas de ficheiros "raiz" ou QOpenSys, o sistema **tem de** estar num estado restrito. Não pode utilizar nenhum dos sistemas de ficheiros durante a conversão. Todos os sistemas de ficheiros UDFS e NFS são desmontados pelo comando CVTDIR e não são montados de novo quando a conversão é concluída. O comando Montar (MOUNT) pode ser utilizado para montar de novo um sistema de ficheiros UDFS ou NFS.

### Converter sistemas de ficheiros definidos pelo utilizador

Quando os UDFSs dos ASPs 1-32 são convertidos, não estão disponíveis para nenhum utilizador do sistema. Para cada um destes ASPs, existe um directório QASPxx no directório /dev. Enquanto o comando CVTDIR está a ser executado, remove o directório QASPxx do espaço de nome para impedir qualquer utilizador de ter acesso aos UDFSs do ASP. Quando o comando CVTDIR acaba de processar todos os objectos, incluindo o directório QASPxx, os objectos são colocados de novo no espaço de nome e tornados disponíveis para os utilizadores do sistema. Os UDFSs do ASP são desmontados pelo comando CVTDIR e não serão montados de novo quando a conversão for concluída. O comando Montar (MOUNT) pode ser utilizado para montar de novo um UDFS.



**Nota:** O directório QASP01 existe em todos os sistemas.

### **Criar perfis de utilizador**

A função de conversão cria perfis de utilizador que são utilizados durante a execução da função de conversão. Estes perfis de utilizador têm o nome QP0FCVxxxx, em que xxxx é um número como, por exemplo, 0001. São utilizados pela função de conversão de modo a serem proprietários de directórios existentes no sistema de ficheiros que são convertidos caso o proprietário original não consiga ser proprietário dos respectivos directórios.

Estes perfis de utilizador serão eliminados quando a conversão for concluída, se possível. A mensagem CPIA08B é enviada se a propriedade de um directório for concedida a um destes perfis de utilizador.

### **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 e válido e é enviada a mensagem CPIA08A para o registo de trabalhos, 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 UTF-16, consulte a homepage de 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 objecto 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 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.

**Alterar o máximo de memória para um perfil de utilizador:** Durante o processamento da conversão de directórios, existe um número de directórios que existe momentaneamente em ambos os formatos e que pertence ao mesmo perfil de utilizador. Se o directório \*TYPE2 não puder ser criado por ter sido

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 e a conversão continua.

**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 e a conversão continua.

## Exemplo: Converter todos os sistemas de ficheiros (pequeno número de objectos)

O sistema A tem cinco conjuntos de memória auxiliar (ASPs) configurados: 1 (o ASP do sistema), 3, 5, 11 e 25. Nenhum destes sistemas de ficheiros foi convertido de directórios TYPE1 em directórios \*TYPE2 no sistema. Suponha que pretende converter todos os sistemas de ficheiros. Como os sistemas de ficheiros não contêm um grande número de objectos, o seu plano consiste em executar todos os passos num só dia.

Para converter directórios em todos os sistemas de ficheiros que tenham um pequeno número de objectos:

1. Coloque o sistema no estado restrito.
2. Escreva RCLSTG SELECT(\*ALL) na linha de comandos.
3. Guarde o sistema utilizando o menu Guardar. Escreva G0 SAVE na linha de comandos e seleccione a opção 21.
4. Escreva CVTDIR OPTION(\*ESTIMATE) FILESYS(\*ALL) FORMAT(\*TYPE2) na linha de comandos.
5. Verifique se existem mensagens de erro apresentadas pela função \*ESTIMATE.
6. Verifique se todos os ASPs têm espaço de memória auxiliar suficiente disponível.
7. Escreva CVTDIR OPTION(\*CONVERT) FILESYS(\*ALL) FORMAT(\*TYPE2) na linha de comandos.

**Nota:** Quando converter todos os sistemas de ficheiros (\*ALL), a mensagem CCAA084 é apresentada e pede-lhe que verifique se os sistemas de ficheiros listados deverão ser convertidos.

8. Verifique se existem mensagens de erro apresentadas pela função \*CONVERT.
9. Retire o sistema do estado restrito.

## Exemplo: Converter todos os sistemas de ficheiros (grande número de objectos)

O sistema B também tem cinco conjuntos de memória auxiliar(ASPs) configurados: 1 (o ASP do sistema), 3, 5, 11 e 25. Nenhum destes sistemas de ficheiros foi convertido de directórios TYPE1 para directórios \*TYPE2 no sistema. Suponha que pretende converter todos os sistemas de ficheiros. Como os sistemas de ficheiros contêm um grande número de objectos, o seu plano consiste em executar todos os passos de conversão em dois fins-de-semana diferentes.

### Fim-de-semana um:

1. Coloque o sistema no estado restrito.
2. Escreva RCLSTG SELECT(\*ALL) na linha de comandos.
3. Guarde o sistema utilizando o menu Guardar. Escreva G0 SAVE na linha de comandos e seleccione a opção 21.
4. Retire o sistema do estado restrito.

### Durante a semana:

5. Escreva CVTDIR OPTION(\*ESTIMATE) FILESYS(\*ALL) FORMAT(\*TYPE2) na linha de comandos.
6. Verifique se existem mensagens de erro apresentadas pela função \*ESTIMATE.

7. Verifique se todos os ASPs têm espaço de memória auxiliar suficiente disponível.

**Fim-de-semana dois:**

8. Coloque o sistema no estado restrito.
9. Escreva `CVTDIR OPTION(*CONVERT) FILESYS(*ALL) FORMAT(*TYPE2)` na linha de comandos.

**Nota:** Quando converter todos os sistemas de ficheiros (\*ALL), a mensagem CPAA084 é apresentada e pede-lhe que verifique se os sistemas de ficheiros listados deverão ser convertidos.

10. Verifique se existem mensagens de erro apresentadas pela função \*CONVERT.
11. Retire o sistema do estado restrito.

## Exemplo: Converter apenas certos ASPs

O sistema C tem seis conjuntos de memória auxiliar (ASPs) configurados: 1 (o ASP do sistema), 2, 4, 8, 10 e 30. Nenhum dos sistemas de ficheiros foi convertido no sistema. A sua intenção é converter apenas os UDFSs nos ASPs 4, 10 e 30.

Para converter os directórios nos UDFSs em certos ASPs:

1. Verifique o formato de directório dos sistemas de ficheiros. Para tal, escreva `CVTDIR OPTION(*CHECK)` na linha de comandos.
2. Coloque o sistema no estado restrito.
3. Escreva `RCLSTG SELECT(*ALL)` na linha de comandos.
4. Guarde o sistema utilizando o menu Guardar. Escreva `G0 SAVE` na linha de comandos e seleccione a opção 21.
5. Retire o sistema do estado restrito.
6. Escreva `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(4) FORMAT(*TYPE2)` na linha de comandos.
7. Verifique se existem mensagens de erro apresentadas pela função \*ESTIMATE.
8. Escreva `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(10) FORMAT(*TYPE2)` na linha de comandos.
9. Verifique se existem mensagens de erro apresentadas pela função \*ESTIMATE.
10. Escreva `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(30) FORMAT(*TYPE2)` na linha de comandos.
11. Verifique se existem mensagens de erro apresentadas pela função \*ESTIMATE.
12. Verifique se todos os ASPs têm espaço de memória auxiliar suficiente disponível.
13. Escreva `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(4) FORMAT(*TYPE2)` na linha de comandos.
14. Verifique se existem mensagens de erro apresentadas pela função \*CONVERT.
15. Escreva `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(10) FORMAT(*TYPE2)` na linha de comandos.
16. Verifique se existem mensagens de erro apresentadas pela função \*CONVERT.
17. Escreva `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(30) FORMAT(*TYPE2)` na linha de comandos.
18. Verifique se existem mensagens de erro apresentadas pela função \*CONVERT.

---

## Nome de caminho

Um **nome de caminho** (também designado por **nomecaminho** nalguns sistemas) 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.

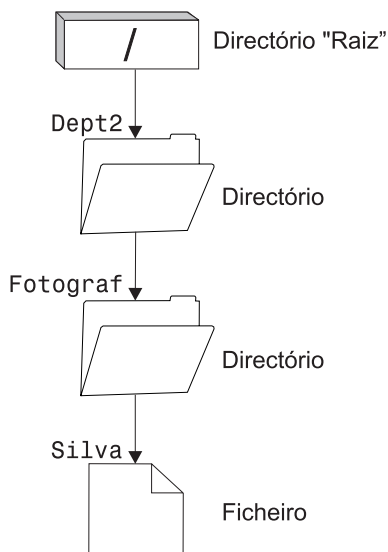


Figura 6. Componentes de um nome de caminho

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

`/Dept2/Fotograf/Silva`

O nome absoluto do caminho 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 no nome é o directório ou objecto no *nível seguinte abaixo* do directório actual.

---

## 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 sob 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 acostumado à 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 do directório inicial de cada utilizador ao ficheiro (consulte “Directório actual e 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: **ligações permanentes** e **ligações simbólicas**.

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

- Ligação permanente
- Ligação simbólica
- Comparação: Ligação permanente e 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 separada 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 é outro directório. Pode haver apenas uma ligação permanente de um directório para outro directório.

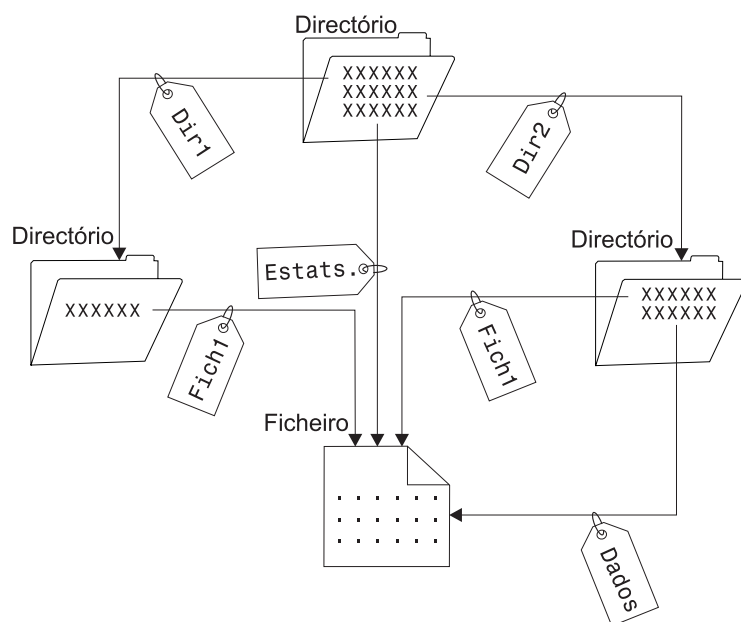


Figura 7. Uma entrada de directório define cada ligação permanente.

As ligações permanentes podem ser removidas sem afectar a existência de um objecto, desde que exista uma ligação permanente, no mínimo, para o 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, um arquivador tem uma ligação permanente a cada documento no arquivador. No entanto, não são permitidas múltiplas ligações permanentes ao *mesmo objecto* no QSYS.LIB, no 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 no sistema 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 quaisquer dos restantes caminhos que seguem 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:

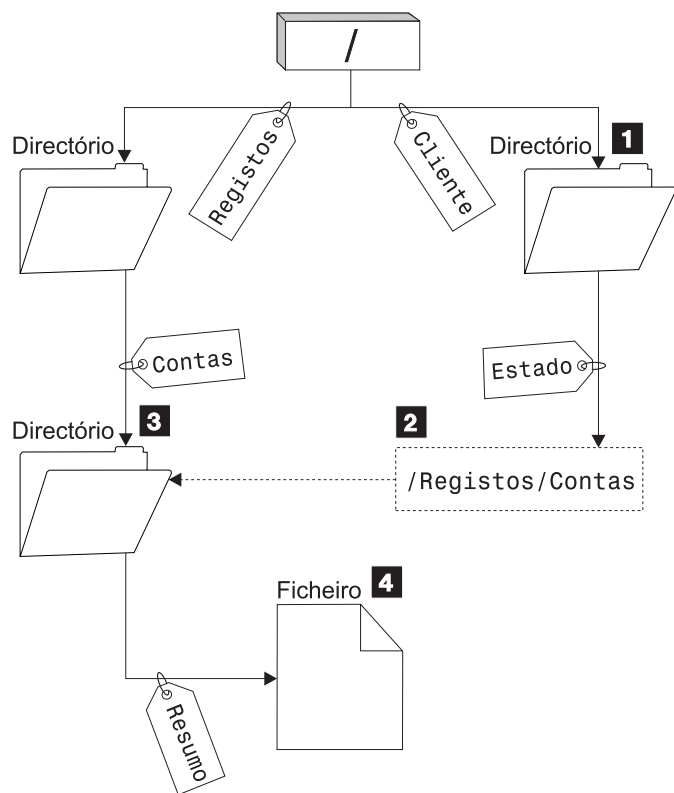


Figura 8. Exemplo de utilização de uma ligação simbólica

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 a 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 em determinado 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 "Comparação: Ligação permanente e ligação simbólica".

## Comparação: Ligação permanente e ligação simbólica

Quando utilizar nomes de caminho em programas, pode optar por utilizar uma ligação permanente ou uma ligação simbólica (consulte "Ligação" na página 18). 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 de objecto	Necessária. Tem de existir um objecto para criar uma ligação permanente a ele.	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 dinâmicos (os atributos são alterados)	Mais lenta. Muitos dos atributos de um objecto são armazenados em cada ligação permanente. Por conseguinte, as alterações a um objecto dinâmico são mais lentas à medida que o número de ligações ao objecto aumenta.	Mais rápida. As alterações ao objecto dinâmico não são afectadas pelas ligações simbólicas.
Objectos estáticos (os atributos não são alterados)	Mais rápida. Para um objecto estático, a resolução do nome é a prioridade da execução primária. 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.

Tabela 1. Comparação entre Ligação Permanente e Ligação Simbólica (continuação)

Item	Ligação Permanente	Ligação Simbólica
Âmbito	Limitada. As ligações permanentes não podem cruzar sistemas de ficheiros.	Ilimitada. As ligações simbólicas podem cruzar sistemas de ficheiros.

## 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 do sistema (SEA) standard, 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), os arquivadores e documentos podem ter qualquer tipo de EA. Alguns arquivadores e documentos poderão ter EAs, outros não. Nos sistemas de ficheiros "raiz" (/), sistemas abertos (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 nenhum 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 77 e "Sistema de Ficheiros Óptico (QOPT)" na página 79 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 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 de sistema standard.
- Para minimizar a possibilidade de conflitos de nomes, utilize uma estrutura de nomenclatura sólida para os EAs. Recomenda-se a forma seguinte:

NomeEmpresaNomeProduto.Nome\_Atributo

---

## 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 continuam iguais. Isto também se aplica à utilização destes sistemas de ficheiros no Servidor de iSeries 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 UTF-16 são subconjuntos da norma ISO 10646. Quando o nome é utilizado, o sistema converte o formato armazenado de caracteres na representação de caracteres adequados na página de códigos a ser utilizada. Os nomes de atributos expandidos associados a cada objecto também são tratados do mesmo modo.

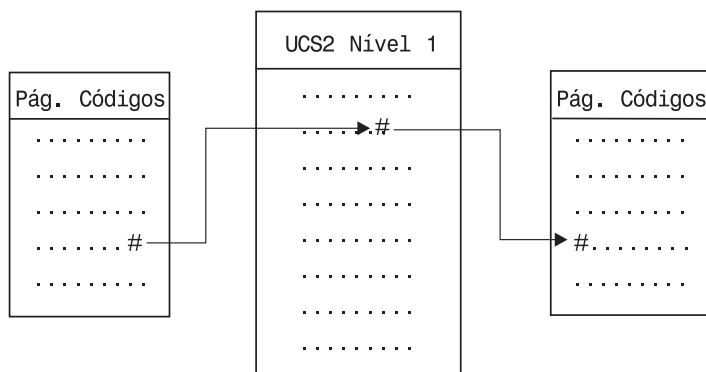


Figura 9. Manter caracteres inalterados independentemente dos esquemas de codificação

Este suporte facilita a interacção com um servidor a partir de dispositivos que utilizam diferentes páginas de códigos. Por exemplo, os utilizadores de PCs 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.



---

## Capítulo 3. Aceder ao sistema de ficheiros integrado utilizando a interface de sistema tradicional

Todas as interfaces de utilizador, tais como menus, comandos e ecrãs, que são utilizadas para trabalhar com bibliotecas, objectos, ficheiros de base de dados, arquivadores 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 conjunto de interfaces de utilizador para o sistema de ficheiros integrado. Estas interfaces podem ser utilizadas em objectos de qualquer sistema de ficheiros que possa ser acedido através dos directórios do sistema de ficheiros integrado.

Pode interagir com os directórios e objectos do sistema de ficheiros integrado a partir do servidor, utilizando menus e ecrãs ou utilizando comandos da linguagem de controlo (CL). Além disso, pode utilizar interfaces de programação 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 de utilizador gráfica utilizada 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:

### Utilizar APIs

As interfaces de programas de aplicação (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.

### Utilizar comandos de CL

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

### Utilizar menus e ecrãs do iSeries

Pode executar operações em ficheiros e outros objectos do sistema de ficheiros integrado utilizando um conjunto de menus e ecrãs fornecidos pelo servidor.

### Utilizar o iSeries Navigator

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

### Utilizar um PC

Se o seu PC estiver ligado a um Servidor de iSeries, pode interagir com os directórios e objectos do sistema de ficheiros integrado como se estivessem armazenados no seu PC.

---

## Executar operações utilizando menus e ecrãs do iSeries

Pode executar operações em ficheiros e outros objectos do 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. Selecciona a opção **Ficheiros, Bibliotecas e Arquivadores** a partir do menu principal iSeries.
4. Selecciona a opção **Sistema de Ficheiros Integrado** no menu Ficheiros, Bibliotecas e Arquivadores.

A partir daqui, pode trabalhar com comandos de Directório, de Objecto ou de Segurança, no sistema de ficheiros integrado, dependendo dos seus requisitos. No entanto, se sabe qual o comando de CL a utilizar, pode escrevê-lo na linha de comandos, no fim do ecrã, e premir **Enter**, ignorando o menu de opções.

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

1. Escreva G0 DATA em qualquer linha de comandos para visualizar o menu Ficheiros, Bibliotecas e Arquivadores.
2. Selecciona a opção Sistema de ficheiros integrado.

Para ver um menu de comandos do Sistema de Ficheiros da Rede, escreva G0 CMDNFS em qualquer linha de comandos. Para ver um menu de comandos do sistema de ficheiros definido pelo utilizador, escreva G0 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
- 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 desinstalar sistemas de ficheiros num cliente

Alguns sistemas de ficheiros não suportam todas estas operações. Para obter informações sobre as restrições de determinado sistema de ficheiros, consulte “Sistemas de ficheiros no sistema de ficheiros integrado” na página 4.

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

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

---

## Executar operações utilizando comandos de CL

Todas as operações que poderá efectuar através dos menus e ecrãs do sistema de ficheiros integrado (consulte “Executar operações utilizando menus e ecrãs do iSeries” na página 25) também 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 que seja acessível através da 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 de Rede e sistemas de ficheiros instalados em geral, consulte “Sistema de ficheiros definido pelo utilizador (UDFS)” na página 67 e “Sistema de Ficheiros da Rede (NFS)” na página 90. Sempre que exista um comando que execute a mesma operação que um comando do OS/2 ou do DOS, é fornecido um nome de comando alternativo para facilitar as tarefas dos utilizadores do OS/2 e do DOS.

Tabela 2. 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 no Registo em Diário. Utiliza registos em diário para aplicar alterações ocorridas desde que um objecto registado no diário foi guardado ou para aplicar alterações até um determinado momento.	
CHGATR	Alterar Atributo. Alterar um atributo para um único objecto, para um grupo de objectos ou para uma árvore de directórios, onde o directório, o conteúdo do mesmo e o conteúdo de todos os subdirectórios nele incluídos alteram o atributo.	
CHGAUD	Alterar Valor de Auditoria. Liga ou desliga a auditoria para 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 a ser utilizado como directório actual.	CD, CHDIR
CHGNFSEXP	Alterar Exportação do Sistema de Ficheiros da Rede. Adiciona árvores de directório 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 Principal. Altera o grupo principal de um utilizador para outro.	
CHKIN	Dar Entrada. Dá entrada de um objecto do qual foi dada saída previamente.	
CHKOUT	Dar Saída. Dá saída de 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 do Ficheiro de dados contínuos. Copia dados de um ficheiro de dados contínuos para um membro de ficheiro de base de dados.	
CPYTOSTMF	Copiar para o 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 sobre a conversão de directórios do sistema de ficheiros integrado do formato *TYPE1 para o formato *TYPE2, ou como executar uma conversão.	
CVTRPCSRC	Converter Origem de RPC. Gera código C de um ficheiro de input escrito na linguagem de Chamada de Procedimento Remoto (RPC).	RPCGEN
DLTUDFS	Eliminar UDFS. Elimina um Ficheiro Definido pelo Utilizador.	

Tabela 2. Comandos do Sistema de Ficheiros Integrado (continuação)

Comando	Descrição	Nome alternativo
DSPAUT	Ver Autoridade. Mostra uma lista de utilizadores autorizados de um objecto e as respectivas autoridades para o objecto.	
DSPCURDIR	Ver Directório Actual. Mostra o nome do directório actual.	
DSPLNK	Ver Ligações de Objecto. Mostra uma lista de objectos num directório e fornece opções para ver informações sobre os objectos.	
DSPF	Ver Ficheiro de Dados Contínuos. Apresenta um ficheiro de dados contínuos ou um ficheiro de base de dados.	
DSPMFSINF	Ver Informações sobre o Sistema de Ficheiros Instalado. Mostra informações sobre um sistema de ficheiros instalado.	STATFS
DSPUDFS	Ver 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 de um objecto ou lista de objectos.	
ENDNFSSVR	Terminar Servidor do Sistema de Ficheiros da Rede. Termina um ou todos os daemons NFS no servidor e no cliente.	
ENDRPCBIND	Terminar Daemon RPC Binder. Termina o daemon RPCBind de Chamada de Procedimento Remoto (RPC).	
MOV	Mover. Move um objecto para um directório diferente	MOVE
RLSIFSLCK	Libertar Bloqueios do Sistema de Ficheiros Integrado. Liberta todos os bloqueios no intervalo de bytes de NFS retidos por um cliente ou num 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 RPC Binder. 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)	
SAV	Guardar. Copia um objecto ou grupo de objectos do sistema para um dispositivo de cópia de segurança	
SNDJRNE <sup>2</sup>	Enviar Registo em Diário. Adiciona registos em diário do utilizador, opcionalmente associados a um objecto registado em diário, a um receptor de registos em diário.	
STRJRN <sup>2</sup>	Iniciar Registo em Diário. Inicia alterações de registos em diário (efectuados 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 NFS no servidor e no cliente.	

Tabela 2. Comandos do Sistema de Ficheiros Integrado (continuação)

Comando	Descrição	Nome alternativo
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 opções 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 Principal. Mostra uma lista de objectos controlados por um grupo principal 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 seguintes tópicos para obter mais informações sobre comandos e restrições de comandos de CL do sistema de ficheiros integrado sobre a utilização destes comandos em determinados sistemas de ficheiros:

- Sistemas de ficheiros no sistema de ficheiros integrado
- Regras de nomes de caminho para comandos de CL e ecrãs
- O tópico CL do iSeries Information Center

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

Quando utilizar um comando ou ecrã do sistema de ficheiros integrado para trabalhar num 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. O termo **objecto** nestas regras refere-se a qualquer directório, ficheiro, ligação ou outro objecto.

- Os nomes dos objectos devem ser únicos 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 do objecto é limitado pelo sistema de ficheiros onde o objecto se encontra e pelo comprimento máximo de uma cadeia de comandos. Os comandos aceitarão nomes do objecto até 255 caracteres de comprimento e nomes de caminho até 5000 caracteres.

Consulte a secção Sistemas de ficheiros no sistema de ficheiros integrado 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:

```
'~/DirUtl/ObjUtl'
```

- 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/DirUtl/ObjUtl'
```

- 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 71 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/PAG.FILE'
```

Consulte “QSYS.LIB de ASP Independente” na página 74 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 “Executar operações utilizando APIs” na página 50).

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

---

## Executar operações utilizando um PC

Se o seu PC estiver ligado a um Servidor de iSeries, 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 Servidor de iSeries 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 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 Servidor de iSeries através do PC, o sistema de ficheiros integrado torna os directórios e objectos do seu 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). O seu PC utiliza clientes de partilha de ficheiros do Windows para obter acesso ao iSeries

NetServer, que é executado no seu Servidor de iSeries.

#### **Transferir ficheiros utilizando o FTP**

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

#### **Trabalhar com ficheiros utilizando o iSeries Navigator**

O iSeries Access inclui o iSeries Navigator, que é ligado a um Servidor de iSeries e torna o sistema de ficheiros integrado disponível para o PC. O iSeries Navigator é a interface gráfica do utilizador para a gestão e administração do seu Servidor de iSeries 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 baseia-se totalmente no NetServer para ter acesso ao sistema de ficheiros integrado. O suporte de NetServer só está disponível para ligações de TCP/IP a um Servidor de iSeries que utilize o OS/400 V4R2 e superiores.

## **Transferir ficheiros utilizando FTP**

O cliente do Protocolo de Transferência de Ficheiros(FTP) permite-lhe transferir ficheiros que se encontram no seu Servidor de iSeries, incluindo os existentes nos sistemas de ficheiros "raiz", QOpenSys, QSYS.LIB, QSYS.LIB de ASP Independente, QOPT e QFileSvr.400. Também lhe permite transferir arquivadores 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 Windows NT Server (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 iSeries Information Center:

- FTP

- Transferir ficheiros com FTP

## Trabalhar com ficheiros utilizando o iSeries NetServer

O Suporte de iSeries para a Vizinhança na Rede do Windows (iSeries NetServer) é uma função do IBM Operating System/400 Versão 5 (OS/400) que permite aos clientes com o Windows terem acesso a caminhos de directório e filas de output partilhados do OS/400. O iSeries NetServer permite que os PCs a utilizar o software do Windows tenham um acesso permanente aos dados e impressoras que são geridos pelo seu iSeries. Os clientes de PC numa rede utilizam simplesmente 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 ter acesso permanente aos dados e impressoras através do iSeries NetServer. Os sistemas de ficheiros Samba (smbfs) podem ser montados a partir do iSeries NetServer de uma forma semelhante à montagem de sistemas de ficheiros de NFS do iSeries. Para obter mais informações, consulte o tópico iSeries NetServer do iSeries Information Center.

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 de iSeries, e, se necessário, alterar uma partilha de ficheiros do iSeries NetServer utilizando o iSeries Navigator.

Para ter acesso 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 **Explore** 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. 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 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/ruir` com o nome `dirruir`, um utilizador poderá ter acesso 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/dirruir`.

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 as considerações para mover objectos para outro sistema de ficheiros para obter mais informações.

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

- iSeries NetServer

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

---

## Mover objectos para outro sistema de ficheiros

Antes de utilizar o sistema de ficheiros integrado para mover objectos entre sistemas de ficheiros, consulte a secção “Considerações sobre a movimentação de objectos para outro 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. Pode ter interesse em 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.

## Considerações sobre a movimentação de objectos 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. Pode ter interesse em 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 mais informações, consulte “Sistemas de ficheiros no sistema de ficheiros integrado” na página 4.

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 podem já não aceder a objectos que são 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 documentos e arquivadores. 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 apenas de 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. O QDLS suporta muitos atributos de "escritório electrónico", mas outros sistemas de ficheiros não. Por conseguinte, o QDLS é um bom local para guardar os documentos de escritório electrónico.

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 do OS/2, estes sistemas de ficheiros podem fornecer um acesso mais rápido a ficheiros.

---

## Directórios fornecidos pelo sistema de ficheiros integrado

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 "Executar operações utilizando APIs" na página 50). Por exemplo, o programa de aplicação pode remover um ficheiro do /tmp utilizando a API unlink().

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

**/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 actual e 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 standard.

**/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 produto de 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 produto de 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.



---

## Capítulo 4. Aceder ao sistema de ficheiros integrado com o iSeries Navigator

O iSeries Navigator é a interface gráfica do utilizador para a gestão e administração dos seus 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 arquivadores

- “Criar um arquivador” na página 40
- “Remover um arquivador” na página 40
- “Dar entrada de um ficheiro”
- “Dar saída de um ficheiro”
- “Configurar permissões para um ficheiro ou arquivador” na página 38
- “Configurar conversão de texto do ficheiro” na página 38
- “Enviar um ficheiro ou arquivador para outro sistema” na página 39
- “Alterar opções para a definição do pacote” na página 39
- “Marcar uma data e uma hora para enviar o ficheiro ou arquivador” na página 40

### Trabalhar com partilhas de ficheiros

- “Criar uma partilha de ficheiros” na página 40
- “Alterar uma partilha de ficheiros” na página 41

### Trabalhar com sistemas de ficheiros definidos pelo utilizador

- “Criar um novo sistema de ficheiros definido pelo utilizador” na página 41
- “Instalar um sistema de ficheiros definido pelo utilizador” na página 41
- “Desinstalar um sistema de ficheiros definido pelo utilizador” na página 42

### Registar objectos em diário

- “Iniciar registo em diário” na página 42
- “Terminar registo em diário” na página 43

---

## 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**.

---

## 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. Selecciono **Propriedades**.
3. Selecciono **Propriedades do Ficheiro -> Utilizar Página**.
4. Faça clique sobre **Dar Saída**.

---

## Configurar permissões para um ficheiro ou arquivador

A adição de permissões a um objecto permite controlar a capacidade de terceiros em manipular esse objecto. Com as permissões, pode permitir que alguns utilizadores só tenham capacidade de visualização dos objectos, enquanto que permite que outros possam de facto editá-los.

Para definir permissões para um ficheiro ou arquivador:

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 ser 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. Selecciono 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, marcando a caixa que se encontra ao lado do quadrado de opção adequado.
10. Faça clique sobre **OK**.

---

## Configurar conversão de texto do ficheiro

Pode configurar a coconversão 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 arquivador para outro sistema

Para enviar um ficheiro ou arquivador 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 arquivador que pretende enviar ser visível.
4. Faça clique com o botão direito do rato sobre o ficheiro ou arquivador e seleccione **Enviar**. O ficheiro ou arquivador aparece na lista Ficheiros e Arquivadores 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 arquivador.
7. Faça clique sobre **OK** para enviar o ficheiro ou arquivador com as actuais definições do pacote e informações da marcação assumidas.

Também pode “Alterar opções para a definição do pacote” ou “Marcar uma data e uma hora para enviar o ficheiro ou arquivador” na página 40.

Quando cria uma definição de pacote, ela é guardada e pode ser reutilizada em qualquer momento para enviar o conjunto de ficheiros e arquivadores 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 mesmo objectos que o primeiro sistema destino.

---

## Alterar opções para a definição do pacote

Uma definição de 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 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 arquivador para outro sistema”.
2. Faça clique sobre o separador **Opções**. As definições assumidas são incluir subarquivadores 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 uma salvaguarda avançada e restaurar opções.
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 arquivador” na página 40.

---

## Marcar uma data e uma hora para enviar o ficheiro ou arquivador

A utilização da função do marcador dá-lhe a flexibilidade para executar o seu trabalho quando for conveniente para si. Para marcar uma data e uma hora para enviar o ficheiro ou arquivador:

1. Execute os passos incluídos em “Enviar um ficheiro ou arquivador para outro sistema” na página 39.
2. Faça clique sobre **Marcar**.
3. Selecciona as opções relativas à altura em que pretende enviar o ficheiro ou arquivador.

---

## Criar um arquivador

Para criar um arquivador:

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 o novo arquivador e seccione **Novo Arquivador**.
5. Escreva um novo nome para o objecto na caixa de diálogo **Novo Arquivador**.
6. Faça clique sobre **OK**.

Quando cria um arquivador no servidor do iSeries, tem de considerar se pretende proteger o novo arquivador (ou objecto) com a gestão de diários. Consulte o tópico Gestão de diários para obter mais informações.

Tópicos relacionados:

- Iniciar registo em diário
- Terminar registo em diário

---

## Remover um arquivador

Para remover um arquivador:

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 arquivador que pretende remover ser visível.
4. Faça clique com o botão direito do rato sobre o ficheiro ou arquivador e seccione **Eliminar**.

---

## 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 o arquivador para o qual pretende criar uma partilha.
5. Faça clique com o botão direito do rato sobre o arquivador para o qual pretende criar uma partilha e seccione **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 o arquivador com a partilha definida que pretende alterar.
5. Faça clique com o botão direito do rato sobre o arquivador com a partilha definida que pretende **Partilhar**.
6. Seleccione **Nova Partilha**.

---

## Criar um novo sistema de ficheiros definido pelo utilizador

Um sistema de ficheiros definido pelo utilizador (UDFS) é um sistema de ficheiros criado por si e para os quais define atributos. Os UDFSs residem em conjuntos de memória auxiliar (ASPs) no sistema.

Para criar um novo sistema de ficheiros definido pelo utilizador (UDFS, User-Defined File System):

1. Expanda o sistema em **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda **Raiz**.
5. Expanda **Dev**.
6. Faça clique sobre o conjunto de memória auxiliar (ASP) que pretende que contenha o novo UDFS.
7. Seleccione **Novo UDFS** no menu **Ficheiro**.
8. Especifique, na caixa de diálogo **Novo Sistema de Ficheiros Definido Pelo Utilizador**, o nome, a descrição (opcional), os valores da auditoria, o formato de ficheiro assumido e se os ficheiros no novo UDFS terão nomes de ficheiro sensíveis a maiúsculas e minúsculas.

---

## Instalar um sistema de ficheiros definido pelo utilizador

Um sistema de ficheiros definido pelo utilizador (UDFS) é um sistema de ficheiros criado por si e para os quais define atributos. Os UDFSs residem em conjuntos de memória auxiliar (ASPs) no sistema. para aceder ou ver os armazenamentos de dados num UDFS, tem de montar o UDFS.

Quando monta um UDFS, ele abrange quaisquer sistemas de ficheiros, directórios ou objectos que existam abaixo do ponto de montagem na hierarquia de arquivadores. Este facto tornará esses sistemas de ficheiros, directórios ou objectos inacessíveis enquanto não desmontar o UDFS. Para assegurar que o acesso a todos os dados dosistema de ficheiros integrado é mantido, monte o UDFS sobre um arquivador vazio. Após o UDFS montado, os ficheiros incluídos no UDFS estarão acessíveis a partir desse arquivador. Quaisquer alterações efectuadas ao arquivador irão alterar o UDFS, em vez do arquivador coberto.

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

Para instalar um sistema de ficheiros definido pelo utilizador (UDFS):

1. Expanda o sistema em **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.

3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda **Raiz**.
5. Expanda **Dev**.
6. Faça clique sobre o conjunto de memória auxiliar (ASP) que contém o UDFS que pretende instalar.
7. Faça clique com o botão direito do rato sobre o UDFS que pretende instalar na coluna **Nome do UDFS** do painel da direita do Operations Navigator.
8. Seleccione **Instalar**.

Caso prefira utilizar a função arrastar e largar, pode instalar um UDFS arrastando-o para um arquivador 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.

---

## Desinstalar um sistema de ficheiros definido pelo utilizador

Quando monta um UDFS, ele abrange quaisquer sistemas de ficheiros, directórios ou objectos existentes sob o ponto de montagem na hierarquia de arquivadores. Este facto tornará esses sistemas de ficheiros, directórios ou objectos inacessíveis enquanto não desmontar o UDFS.

Para desinstalar um sistema de ficheiros definido pelo utilizador (UDFS):

1. Expanda o sistema em **Operations Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Expanda **Sistema de Ficheiros Integrado**.
4. Expanda **Raiz**.
5. Expanda **Dev**.
6. Faça clique sobre o conjunto de memória auxiliar (ASP) que contém o UDFS que pretende desinstalar.
7. Faça clique com o botão direito do rato sobre o UDFS que pretende desmontar na coluna **Nome do UDFS** do painel da direita do iSeries Navigator.
8. Seleccione **Desinstalar**.

---

## Iniciar registo em diário

O principal objectivo do registo em diário é permitir-lhe recuperar as alterações a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado.

Para iniciar o registo em diário num objecto:

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 obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte Journal management no iSeries Information Center.

---

## Terminar registo em diário

O principal objectivo do registo em diário é permitir-lhe recuperar as alterações 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:

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 obter informações mais detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte Journal management no iSeries Information Center.



---

## Capítulo 5. Programar suporte para o sistema de ficheiros integrado

A adição do sistema de ficheiros integrado ao Servidor de iSeries 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 programação de aplicações (APIs) em linguagem C, 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 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

---

### 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 contínuo; 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 STRDDLML 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 Command language (CL) do iSeries Information Center.

## Copiar dados utilizando APIs

Se pretender 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 e ler e escrever dados nele. Consulte o tópico Integrated File System APIs do iSeries Information Center para obter mais informações.

## Copiar dados utilizando a função de transferência de dados

As aplicações de transferência de dados do iSeries Access 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 e a utilização de recursos de PC e do Servidor de iSeries e comunicações entre os dois.

Se tiver instalado o iSeries Access no PC e o 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. Selecciona **Transferência de Dados do Servidor de iSeries** a partir da janela iSeries Access para Windows.
4. Selecciona o servidor a partir do qual pretende efectuar a transferência.
5. Selecciona 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 seleccione a opção de menu **Ficheiro** para guardar o pedido de transferência. A aplicação Transferência de Dados para o Servidor de iSeries cria um ficheiro .DTT ou .TFR. A aplicação Transferência de Dados do Servidor de iSeries cria um ficheiro .DTF ou .TTO. No directório iSeries Access, 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 MEUFICH.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 Servidor de iSeries** 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 Servidor de iSeries** 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 Servidor de iSeries 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 Servidor de iSeries 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 Servidor de iSeries cria um ficheiro .DTT ou .TFR. A aplicação Transferência de Dados do Servidor de iSeries cria um ficheiro .DTF ou .TTO. No directório iSeries Access, 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 (faz parte do pacote Microsoft Plus) para especificar o programa a executar (por exemplo, RTOPCB MEUFICH.TT0) 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. Selecciona **Transferência de Dados para o Servidor de iSeries** a partir da janela iSeries Access para Windows.
4. Abra o menu **Ferramentas** da aplicação Transferência de Dados para o Servidor de iSeries.
5. Selecciona **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 Servidor de iSeries 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 Servidor de iSeries 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 Servidor de iSeries 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 Servidor de iSeries. 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

As interfaces de programas de aplicação que executam operações em directórios e ficheiros de dados contínuos 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 (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.

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 3. 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
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
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

Tabela 3. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
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
getwpmam()	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
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

Tabela 3. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
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 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)
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 do directório actual (utilizando nome de caminho activado para NLS)
QlgGetPathFromFileID()	Obter nome 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 (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)



Tabela 3. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
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)
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
Qp0ICvtPathToSYSObjName()	Processar nome de caminho do sistema de ficheiros integrado para Nome de Objecto de QSYS
Qp0IFLOP()	Executar operações diversas em objectos

Tabela 3. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
Qp0IGetAttr()	Obter atributos de sistema para um objecto
Qp0IGetPathFromFileID()	Obter nome de caminho do objecto a partir do respectivo ID do ficheiro
Qp0IOpen()	Abrir ficheiro com o nome de caminho activado para NLS
Qp0IProcessSubtree()	Processar directórios ou objectos numa árvore de directórios
Qp0IRenameKeep()	Mudar o nome do ficheiro ou directório, manter <i>novo</i> , se existir
Qp0IRenameUnlink()	Mudar o nome do ficheiro ou directório, desligar <i>novo</i> se existir
QP0LROR()	Obter referências a objectos
Qp0ISaveStgFree()	Guardar dados de objectos e libertar a respectiva memória
Qp0ISetAttr()	Definir atributos de sistema para um objecto
Qp0IUnlink()	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 do directório. Pode ser definido para ter a semântica Qp0IRenameKeep() ou Qp0IRenameUnlink().
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



Tabela 3. APIs do Sistema de Ficheiros Integrado (continuação)

Função	Descrição
utime()	Definir horas de acesso e de modificação do ficheiro
write()	Escrever no ficheiro
writenv()	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 “Sistemas de ficheiros no sistema de ficheiros integrado” na página 4. Para obter um exemplo de um programa que utiliza as funções C do sistema de ficheiros integrado, consulte o Apêndice B, “Programa exemplo que utiliza funções C do sistema de ficheiros integrado” na página 103.

Consulte os seguintes tópicos para obter mais informações sobre APIs do sistema de ficheiros integrado:

- Funções de ILE C/400
- Suporte de ficheiros grandes para APIs
- Regras de nomes de caminhos para APIs
- Descritor de ficheiros
- Segurança
- O tópico Application programming interfaces (APIs) do iSeries Information Center

## Funções de ILE C/400

O ILE C/400 fornece as funções C standard 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.

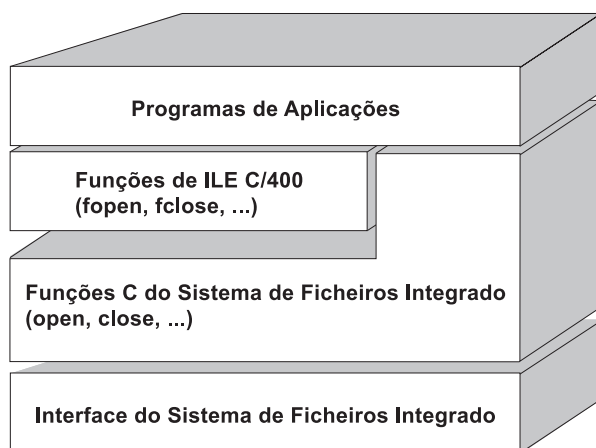




Figura 10. As funções de ILE C/400 utilizam as funções de I/O de sequência do sistema de ficheiros integrado.

Para obter mais informações sobre a utilização de funções de ILE C/400 com I/O de sequência 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 para APIs

As APIs do sistema de ficheiros integrado foram melhoradas para permitir que as suas aplicações armazenem e manipulem ficheiros muito grandes. O ficheiro de dados contínuos do sistema de ficheiros integrado permite tamanhos de ficheiros de dados contínuos de até 256 gigabytes nos sistemas de ficheiros "raiz" (/), Open Systems File System (QOpenSys), e 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 iSeries Information Center.

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()` API 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 com um 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 61 para obter os limites de cada sistema de ficheiros.

- / 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 Servidor de iSeries, 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 71 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/PAG.FILE
```

Consulte “QSYS.LIB de ASP Independente” na página 74 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 29), 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.

## 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 **descritor de ficheiro**. Um descritor de ficheiros é um número inteiro não negativo que deve ser único em cada trabalho. O trabalho utiliza um descritor de ficheiros para identificar um ficheiro aberto durante a execução de operações no ficheiro. O descritor de ficheiros é representado pela variável *filides* nas funções C utilizadas no sistema de ficheiros integrado e pela variável *descriptor* nas funções C utilizadas em sockets.

Cada descritor 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 descritor do ficheiro, mas um descritor de ficheiro pode referir-se apenas a uma descrição de ficheiro aberto.

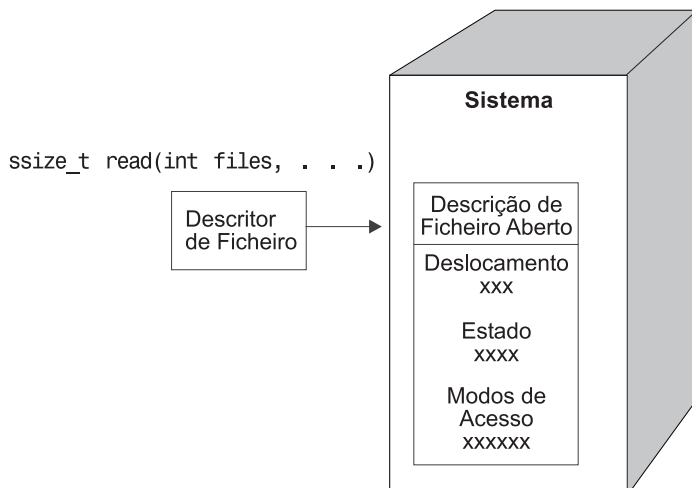


Figura 11. Descritor de ficheiros e descrição de ficheiro aberto

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 descritor de ficheiro.

Quando utilizar os sistemas de ficheiros “raiz” (/), QOpenSys ou definido 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 descritor de ficheiros entre trabalhos. Para obter uma descrição destas funções, consulte Programação de Sockets ou o tópico Sockets APIs do iSeries Information Center.

## 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. A 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 o manual Security — Reference




---

## Suporte de sockets

Se a aplicação estiver a utilizar os sistemas de ficheiros “raiz” (/), QOpenSys ou definido 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 `bink()` 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 na secção Sockets Programming do iSeries Information Center.

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 standard (consulte “Continuidade de nomes” na página 23). 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, as mesmas considerações também se aplicam.

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

---

## Conversão de dados

Quando acede a ficheiros através da 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 ter 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 sistema de ficheiros integrado `open()`.

## Capítulo 6. Sistemas de ficheiros no sistema de ficheiros integrado

Os sistemas de ficheiros no sistema de ficheiros integrado são:

- “raiz”
- QOpenSys
- UDFS
- QSYS.LIB
- QSYS.LIB de ASP Independente
- QDLS
- QOPT
- QNetWare
- QNTC
- QFileSvr.400
- NFS

Para obter uma descrição geral de cada sistema de ficheiros, consulte Comparação entre sistemas de ficheiros.

### Comparação entre sistemas de ficheiros

A Tabela 4 e a Tabela 5 na página 63 resumem as funções e limitações de cada sistema de ficheiros.

Tabela 4. 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
Integrado no 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 processador de I/O do servidor de ficheiros <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	64KB
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



Tabela 4. Resumo dos Sistemas de Ficheiros (Parte 1 de 2) (continuação)

Capacidade	/ (“raiz”)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
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

**Notas:**

1. O processador de I/O do servidor de ficheiros é hardware utilizado pelo Servidor de LAN.
2. Quando acedido através do servidor de ficheiros do OS/400.
3. Quando acedido através de um PC de cliente do Servidor de LAN. O acesso utilizando as APIs do iSeries é comparativamente lento.
4. 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 71 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 74 para obter detalhes.
5. Consulte “Sistema de Ficheiros de Serviços da Biblioteca de Documentos (QDLS)” na página 77 para obter detalhes.
6. 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 71 para obter mais detalhes.
7. 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 77 para obter detalhes.
8. Assumindo um nome absoluto de caminho que começa por / seguido do nome do sistema de ficheiros (tal como /QDLS...).
9. 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.
10. Na prática, os níveis de directórios são limitados pelos limites de espaço do programa e do sistema.
11. Excepto um directório, que pode ter apenas uma ligação a outro directório.
12. 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.
13. 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.
14. Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP Independente podem suportar 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.
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.
16. Os dados desta coluna referem-se a ambos os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP Independente.

**Abreviaturas**

car = caracteres

B = bytes    KB = quilobytes    MB = megabytes    GB = gigabytes



Tabela 5. 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
Tipo de ficheiro	Dados Contínuos	Dados Contínuos	Dados Contínuos	Dados Contínuos	Dados Contínuos
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 Servidor de PC Integrado (FSIOP) <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	Não	Sim	Não	Não
Suporta registo em diário de objectos	Não	Não	Sim	Não	Não

Tabela 5. 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 processador de I/O do servidor de ficheiros é hardware utilizado pelo Servidor de LAN.</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 79 para obter detalhes.</li> <li>5. Assumindo um nome absoluto de caminho que começa com 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> </ol>					
<b>Abreviaturas</b>					
car = caracteres					
B = bytes    KB = kilobytes    MB = megabytes    GB = gigabytes					

## 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 Disk Operating System (DOS) e OS/2.

Para além disso:

- Está optimizado para input/output de ficheiros de dados contínuos.
- Suporta múltiplas ligações permanentes e ligações simbólicas.
- Suporta sockets locais.
- Suporta objectos \*OOPOOL.
- Suporta interfaces de programação de aplicações (APIs) com protecção para módulos.
- Suporta objectos \*FIFO.
- Suporta os objectos /dev/null e /dev/zero \*CHRSF, bem como outros objectos \*CHRSF.
- Suporta o registo em diário de alterações a objectos.

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 que utilize o servidor de ficheiros do OS/400 ou os comandos do sistema de ficheiros integrado, ecrãs de utilizador e APIs de linguagem C.

### 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

- Os componentes dos nomes de caminho após o nome da partilha seguem as regras de nomenclatura de caminhos do Windows NT. Os nomes dos directórios e o nome do objecto combinados podem ocupar o comprimento total do caminho, ou seja 255 caracteres.
- 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 23). Consulte os 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 18.

### Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros "raiz" (/)

Todos os comandos incluídos em "Executar operações utilizando comandos de CL" na página 26 e os ecrãs descritos em "Executar operações utilizando menus e ecrãs do iSeries" na página 25 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 da linguagem C incluídas em "Executar operações utilizando APIs" na página 50 podem ser utilizadas no sistema de ficheiros "raiz" (/) com protecção para 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 o Capítulo 7, "Suporte de registo em diário para objectos do sistema de ficheiros integrado" na página 93.

---

## 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 otimizado para input/output de ficheiros de dados contínuos.
- Suporta múltiplas ligações permanentes e ligações simbólicas.
- Suporta nomes sensíveis a maiúsculas e minúsculas.
- Suporta sockets locais.
- Suporta objectos \*OOPOOL.
- Suporta APIs com protecção contra módulos.
- Suporta objectos \*FIFO.
- Suporta o registo em diário de alterações a objectos.

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.

Para obter mais informações sobre conversões de directórios \*TYPE 1 para \*TYPE2 e as restrições do sistema de ficheiros QOpenSys, consulte Directórios \*TYPE2.

## Utilizar o QOpenSys

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

### 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 23). Consulte os directórios 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 18 para obter uma descrição de ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys**

Todos os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 e os ecrãs descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25 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 funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 podem funcionar no sistema de ficheiros QOpenSys com protecção contra 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 objectos do sistema de ficheiros QOpenSys, consulte o Capítulo 7, “Suporte de registo em diário para objectos do sistema de ficheiros integrado” na página 93.

---

## **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 otimizados para input/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.

Pode criar múltiplos UDFSs atribuindo a cada um nome único. 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.

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

- Conceitos de UDFS
- Utilizar o UDFS através do 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 SOM.

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*, *accão* 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/accão 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 montar um UDFS por cima de outro num ASP independente.

Para obter mais informações sobre a instalação de sistemas de ficheiros, consulte Suporte do Sistema de

Ficheiros de Rede do OS/400  .

## 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 os comandos, os ecrãs do utilizador e as APIs do servidor de ficheiros do OS/400 ou do sistema de ficheiros integrado. 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

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 irá corresponder 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 irá distinguir entre maiúsculas e minúsculas durante a procura de nomes. Deste modo, o servidor reconheceria /home/CARLA



e /HOME/carla como o mesmo directório e não como dois directórios separados. 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 23). Consulte os Directórios \*TYPE2 para obter mais informações sobre os formatos de directório.

### **Ligações num UDFS do sistema de ficheiros integrado**

Os objectos de um UDFS permitem estabelecer múltiplas ligações permanentes ao mesmo objecto, suportando totalmente o estabelecimento de ligações simbólicas. Uma ligação simbólica pode criar uma ligação de um UDFS a um objecto noutra sistema de ficheiros.

Consulte “Ligação” na página 18 para obter uma descrição de ligações.

### **Utilizar comandos do sistema de ficheiros integrado num UDFS**

Todos os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 e os ecrãs descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25 podem funcionar num sistema de ficheiros descrito 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 6. 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 clientes locais.
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	Ver UDFS - Mostra informações sobre um sistema de ficheiros definido pelo utilizador.
MOUNT	Instalar sistema de ficheiros - Coloca sistemas de ficheiros exportados do servidor remoto em directórios de clientes locais. Este comando é uma alternativa ao comando ADDMFS.
RMVMFS	Remover Sistema de Ficheiros Instalado - Remove sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local.
UNMOUNT	Desinstalar sistema de ficheiros - Remove 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 funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 podem funcionar num sistema de ficheiros definido pelo utilizador.

**Nota:** Tem de instalar um UDFS para poder executar comandos do sistema de ficheiros integrado 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 a partir de um cliente do Windows NT.

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

- “Criar um novo sistema de ficheiros definido pelo utilizador” na página 41.
- “Instalar um sistema de ficheiros definido pelo utilizador” na página 41.
- “Desinstalar um sistema de ficheiros definido pelo utilizador” na página 42.

### 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 a sensibilidade a maiúsculas e minúsculas.

### 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 o Sistema de Ficheiros Instalado (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 montar 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 Montado (RMVMFS) ou UNMOUNT tornarão um sistema de ficheiros montado 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 montado. 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 montar 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 desmontado.

## Registar em diário alterações a objectos de um 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 o Capítulo 7, "Suporte de registo em diário para objectos do sistema de ficheiros integrado" na página 93.

---

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

O sistema de ficheiros QSYS.LIB suporta a estrutura de bibliotecas do Servidor de iSeries. Este sistema de ficheiros fornece-lhe acesso a ficheiros de base de dados e a todos os outros tipos de objectos do Servidor de iSeries 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 Servidor de iSeries 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 gerir 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 mais 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 os comandos do sistema de ficheiros integrado, ecrãs de utilizador e APIs de linguagem C. 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**

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 iSeries Information Center.)
- 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/output de sequência para objectos de espaço de utilizador. Por exemplo, um programa pode escrever dados contínuos 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 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/PAG.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 iSeries Information Center.

## 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 18 para obter uma descrição de ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB**

Os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 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 descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

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

As funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 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 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 Servidor de iSeries 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 Servidor de iSeries 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 Servidor de iSeries 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 Servidor de iSeries
- 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 os comandos, ecrãs de utilizador e APIs de linguagem C 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 QPWFSERVER no sistema de ficheiros QSYS.LIB de ASP Independente**

A QPWFSERVER 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 do tratamento 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 iSeries Information Center.)
- 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/output de sequência para objectos espaços do 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 de I/O sequência para objectos 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/PAG.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 noCCSID 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 iSeries Information Center.

## 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 18 para obter uma descrição de ligações.

## Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QSYS.LIB de ASP Independente

Os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 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 simbólicaa 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 do 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 ser 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 descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

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

As funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 podem funcionar no sistema de ficheiros QSYS.LIB de ASP Independente, excepto quanto ao seguinte:

- 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 no 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 arquivadores. Permite o acesso a documentos e arquivadores.

Para além disso:

- Suporta arquivadores 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 os comandos, ecrãs de utilizador e APIs de linguagem C 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.

### **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 dos 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 que o sistema de ficheiros integrado se baseia no modelo do programa do Integrated Language Environment (ILE), o HFS baseia-se, por seu lado, no modelo do 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*).

Certifique-se de 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. Se `/` e `QDLS` forem incluídos nos 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, 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 um arquivador e os objectos da biblioteca de documentos como se tratasse de uma ligação entre o arquivador e cada objecto no arquivador. 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 18 para obter uma descrição das ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QDLS**

Os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 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`, `ENDJRN`, `SNDJRNE` e `STRJRN` não são suportados.

Aplicam-se as mesmas restrições aos ecrãs do utilizador descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

### **Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QDLS**

As funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 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 hierárquica de directórios semelhante à dos sistemas operativos de PC, tais como DOS e OS/2.
- Está optimizado para input/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 da interface do sistema de ficheiros integrado.

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

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


Para mais detalhes, consulte o manual 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 do 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. Certifique-se de 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 no iSeries Information Center ou na 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 com 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 simplesmente especificar /QOPT no nome de 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 do nome de caminho no sistema de ficheiros QOPT, consulte a secção “Path Name Rules” do manual .

### **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 do 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 18 para obter uma descrição de ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QOPT**

A maior parte dos comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 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 descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

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

- ADDLNK
- APYJRNCHG
- CHKIN
- CHKOUT
- ENDJRN

- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

### **Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOPT**

Todas as APIs da linguagem C incluídas em “Executar operações utilizando APIs” na página 50 podem funcionar no sistema de ficheiros “raiz” (/) com protecção contra módulos.

- 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, 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:

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

### **Montar sistemas de ficheiros de NetWare**

Os sistemas de ficheiros de 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 de 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 de 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 de NetWare MAP. Os caminhos NDS podem ser utilizados para especificar um caminho para o volume NetWare mas não podem ser 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
```

As extensões .TRE, .C, .0, .OU, e .CN são utilizadas para representar árvores NDS, países, organizações, unidades organizacionais e nomes comuns, respectivamente. Se um volume da Novell Netware for acedido pelo caminho NDS através de um objecto volume ou de um nome alternativo para 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 os comandos, ecrãs de utilizador e APIs do sistema de ficheiros integrado. Deve ter em atenção as considerações, limitações e dependências que se seguem.

### Autoridades e propriedade no sistema de ficheiros QNetWare

Os ficheiros e os directórios do QNetWare são armazenados e geridos pelos servidores da Novell NetWare. Quando utilizar comandos e APIs para obter ou definir as autoridades dos proprietários ou dos utilizadores, QNetWare definirá utilizadores da NetWare como utilizadores do servidor iSeries com base no nome de um utilizador. Se o nome da 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 as autoridades são transferidas de e para o servidor, serão definidas 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 ao 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 da Novell NetWare, os ficheiros e directório aparecerão todos 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 “Executar operações utilizando comandos de CL” na página 26 podem funcionar no sistema de ficheiros QNetWare, com as seguintes excepções:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRN
- 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 funções da linguagem C incluídas em “Executar operações utilizando APIs” na página 50 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 Windows NT Server (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 superior, ou um servidor autónomo. Permite às aplicações do servidor iSeries utilizar os mesmos dados que os clientes do Windows NT. Armazena dados em ficheiros de dados contínuos.

O sistema de ficheiros QNTC faz parte do sistema operativo OS/400 base. Para utilizar o sistema de ficheiros QNTC, tem de ter o TCP/IP Connectivity Utilities for iSeries 400 (part number: 5769-TC1) instalado. 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 servidor de ficheiros do OS/400 ou os comandos, ecrãs do utilizador e 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 considerações e limitações que se seguem.

### 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 de servidor do NT é administrada a partir do Windows NT Server. 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 com 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 Windows NT Server, no nome da partilha, nos nomes do directório e subdirectório 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 estão armazenados no 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 Tabela 2 na página 27) ou a API mkdir() (consulte “Executar operações utilizando APIs” na página 50) para adicionar servidores Windows NT 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 18 para obter uma descrição de ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNTC**

Os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 podem funcionar no sistema de ficheiros QNTC, com as seguintes excepções:

ADDLNK  
 APYJRNCHG  
 CHGOWN  
 CHGAUT  
 CHGPGP  
 CHKIN  
 CHKOUT  
 DSPAUT  
 ENDJRN  
 RST  
 SAV  
 SNDJRNE  
 STRJRN  
 WRKAUT  
 WRKOBJOWN  
 WRKOBJPGP

Aplicam-se as mesmas restrições aos ecrãs do utilizador que estão descritas em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

### **Utilizar o comando MKDIR no sistema de ficheiros QNTC**

Utilize o comando Criar Directório (MKDIR) para adicionar um directório do servidor ao directório /QNTC. Todos os servidores funcionais do Windows NT na sub-rede local são criados automaticamente. Os servidores do Windows NT que estiverem fora da sub-rede local 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'

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

**Nota:** Se utilizar uma API mkdir() ou o comando de CL MKDIR para adicionar directórios à estrutura de directórios, estes não são permanentes 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 funções da linguagem C incluídas na “Executar operações utilizando APIs” na página 50 podem funcionar no sistema de ficheiros QNTC, excepto as seguintes:

- As funções chmod(), fchmod(), utime() e umask() não terão qualquer efeito nos 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()

---

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

O sistema de ficheiros do servidor de ficheiros do OS/400 fornece acesso transparente a outros sistemas de ficheiros que residam nos 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 os 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.

## 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 do 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 diversas 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 dos nomes são convertidos no formato de UCS2 Nível 1 quando os nomes são armazenados (consulte “Continuidade de nomes” na página 23).

## 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 o manual 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 com o 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 atribuído. 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 serviço 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 de TCP com o 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 QPWFSEVSO têm o controlo da ligação. Cada trabalho iniciado previamente é executado com o respectivo perfil do utilizador.

3. Se o nome da localização remota não for atribuído a um endereço de IP, o directório de primeiro nível é considerado como um nome de SNA LU 6.2. 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 o Kerberos, o Kerberos pode ser utilizado para autenticar um sistema de ficheiros que resida num servidor iSeries destino. Se a autenticação do Kerberos falhar, o ID de utilizador e a palavra-passe podem ser utilizados para verificar o acesso.

**Nota:** Se o bilhete de concessão de bilhetes ou o bilhete 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 o Kerberos, consulte o tópico Network authentication service do iSeries Information Center.

- Para ter acesso a um sistema de ficheiros residente num Servidor de iSeries destino, terá de ter um ID de utilizador e uma palavra-passe no sistema 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 está 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 18 para obter uma descrição de ligações.

### **Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros servidor de ficheiros do OS/400**

Os comandos incluídos em “Executar operações utilizando comandos de CL” na página 26 podem funcionar no sistema de ficheiros QFileSvr.400, com as seguintes excepções:

ADDLNK  
 APYJRNCHG  
 CHGAUT  
 CHGOWN  
 DSPAUT  
 ENDJRN  
 RST  
 SAV  
 SNDJRNE  
 STRJRN  
 WRKOBJOWN  
 WRKOBJPGP

Aplicam-se as mesmas restrições aos ecrãs do utilizador descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25.

### **Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros servidor de ficheiros do OS/400**

As funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 podem funcionar no sistema de ficheiros QFileSvr.400, com as seguintes excepções:

chown()  
 fchown()  
 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 o 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 acessado através da interface do sistema de ficheiros integrado e tem as seguintes considerações e limitações.

### 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 de 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 irá funcionar 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 Sistema de Ficheiros da Rede irá funcionar e as características que terá:

1. O utilizador instala um sistema de ficheiros num cliente a partir de um servidor iSeries.
2. O utilizador instala um sistema de ficheiros num cliente a partir de um servidor UNIX.
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 Servidor de iSeries. No entanto, ambas as características do Sistema de Ficheiros de Rede e do sistema de ficheiros que está a ser assistido têm de ser tomados 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 de 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.

Na segunda opção, é importante perceber que qualquer sistema de ficheiros instalado a partir de um servidor UNIX terá um comportamento muito parecido com o 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 66.

Na terceira opção, 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 para múltiplas ligações permanentes e simbólicas dependem por completo do sistema de ficheiros que estiver a ser instalado com o NFS.

Consulte “Ligação” na página 18 para obter uma descrição de ligações.

### **Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros de rede**

Todos os comandos que estão incluídos em “Executar operações utilizando comandos de CL” na página 26 e os ecrãs descritos em “Executar operações utilizando menus e ecrãs do iSeries” na página 25 podem funcionar no Sistema de Ficheiros de Rede, com excepção dos seguintes:

- APYJRNCHG
- 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 de comandos e ecrãs directamente relacionados com o Sistema de Ficheiros da Rede, consulte

o manual OS/400 Network File System Support  .

*Tabela 7. Comandos de CL do Sistema de Ficheiros da Rede*

<b>Comando</b>	<b>Descrição</b>
ADDMFS	Adicionar Sistema de Ficheiros Instalado - Coloca sistemas de ficheiros exportados do servidor remoto em directórios de clientes locais.

Tabela 7. Comandos de CL do Sistema de Ficheiros da Rede (continuação)


Comando	Descrição
CHGNFSEXP	Alterar Exportação do Sistema de Ficheiros da Rede - Adiciona ou remove árvores de directórios da tabela de exportação dos sistemas de ficheiros exportados para clientes do Sistema de Ficheiros da Rede.
DSPMF5INF	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 sistema de ficheiros - Adiciona ou remove árvores de directórios da tabela de exportação dos sistemas de ficheiros exportados para clientes do Sistema de Ficheiros da Rede.
MOUNT	Instalar sistema de ficheiros - Coloca sistemas de ficheiros exportados do servidor remoto em directórios de clientes locais. Este comando é uma alternativa ao comando ADDMFS.
RLSIFSLCK	Libertar Bloqueios do Sistema de Ficheiros Integrado - Liberta todos os bloqueios de intervalos de bytes do Sistema de Ficheiros da Rede retidos por um cliente ou num objecto.
RMVMFS	Remover Sistema de Ficheiros Instalado - Remove 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 sistema de ficheiros - Remove 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 funções de linguagem C incluídas em “Executar operações utilizando APIs” na página 50 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 especificamente relacionadas com o Sistema de Ficheiros de Rede, consulte Suporte do Sistema de Ficheiros de Rede do OS/400  .

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

---

## Capítulo 7. Suporte de registo em diário para objectos do sistema de ficheiros integrado

O principal objectivo do registo em diário é permitir-lhe recuperar as alterações a um objecto que tenham ocorrido desde a última vez que o objecto foi guardado.

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.

Os tópicos que se seguem introduzem o suporte de registo em diário para objectos sistema de ficheiros integrado:

- “Gestão de registos em diário”
- “Objectos que deve registar em diário”
- “Objectos do sistema de ficheiros integrado registados em diário” na página 94
- “Operações registadas em diário” na página 95
- “Considerações especiais para entradas de diário” na página 95

Para obter informações detalhadas sobre o registo em diário de objectos do sistema de ficheiros integrado, consulte o tópico Journal management no iSeries Information Center.

---

### 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
- 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 contínuo, 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 informações sobre o 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.

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 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 "/MiEmpresa" que correspondam 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:** Se terminar o registo em diário de um objecto e, em seguida, mudar o nome a esse 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 ligado.

- 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 APIs write() e writev() é de 2 gigabytes-1. O tamanho máximo de uma entrada de diário, se RCVSIZOPT (\*MAXOPT2) 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 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 *Journal management* no iSeries Information Center.

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 de sistema de ficheiros integrado, consulte o *Journal code finder* no iSeries Information Center.

---

## 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 de dados
- Truncatura/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
- Terminar registo em diário
- Iniciar o comando Aplicar Alterações Registadas (APYJRNCHG)
- Terminar o comando Aplicar Alterações Registadas (APYJRNCHG)
- Guardar
- Restaurar

Para obter informações mais detalhadas sobre o registo em diário de objectos de sistema de ficheiros integrado, consulte o tópico *Journal management* no iSeries Information Center. Para obter uma lista completa de todas as entradas de registo depositadas para objectos de sistema de ficheiros integrado, consulte o *Journal code finder* no tópico **Systems management**.

---

## Considerações especiais para entradas de diário

Muitas operações de sistema de ficheiros integrado registados 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 Remove

Alterações (Código C do Registo, Tipo RB) no ciclo de controlo de consolidação não correspondem a operações falhadas e as entradas de registo incluídas nas mesmas não devem reproduzidas nem replicadas.

As entradas de 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 registo em diário herdado como Sim)
- OA — Alteração de Autoridades
- OG — Alteração do Grupo Principal 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 e/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.



---

## Apêndice A. 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 estão presentemente disponíveis no servidor iSeries são TCP e UDP.

O desenvolvimento de aplicações distribuídas numa rede é uma tarefa fácil quando o RPC é utilizado. Os alvos principais são aplicações que se destinam mais à distribuição da interface do utilizador ou à obtenção de dados.

---

### 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
freenetconfignt()	Liberta a estrutura do netconfig que é devolvida da chamada à função getnetconfignt()
getnetconfig()	Devolve o indicador ao registo actual no ficheiro netconfig e incrementa o respectivo indicador para o registo seguinte
getnetconfignt()	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



API	Descrição
netdir_sperror()	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 arbitrárias, independentemente das diferentes ou convenções de esquema de estruturas ou ordens de bytes do respectivo sistema central.

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.
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

<b>API</b>	<b>Descrição</b>
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 ponteiro 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
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).

<b>API</b>	<b>Descrição</b>
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 com 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.

### 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 selecione 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

<b>API</b>	<b>Descrição</b>
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.

<b>API</b>	<b>Descrição</b>
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_getrpccaller()	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
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



---

## Apêndice B. Programa exemplo que utiliza 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 em 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.

Estas informações de renúncia referem-se a exemplos de código.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
```

```
#define BUFFER_SIZE      2048
#define NEW_DIRECTORY    "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 cadeia 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;
}
```

---

## Apêndice C. Exemplo de código RPG do sistema de ficheiros integrado

A ligação Fragmentos de Código (Code Snippets)  contém um exemplo de código RPG do sistema de ficheiros integrado. Para ver este exemplo, execute os seguintes passos:

1. Seleccione **ILE RPG Source** na lista pendente da categoria Search.
2. Faça clique sobre **Search**.
3. Percorra a lista até ser apresentada a opção **Using IFS from RPG**.
4. Faça clique sobre **Code for using IFS from RPG**.








Estas informações de renúncia referem-se a exemplos de código.



---

## Bibliografia

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 Globalização do Information Center do iSeries explica os conceitos de suporte de idioma nacional (NLS), como, por exemplo, conjunto de caracteres e página de códigos, e fornece informações necessárias para avaliar, planear e utilizar as capacidades de múltiplos idiomas e de NLS do servidor iSeries.
- O tópico APIs da categoria **Programming** do iSeries Information Center fornece uma descrição de cada API do OS/400, incluindo as APIs de 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 de sistema de ficheiros integrado, como uma unidade de trabalho lógica.
- OS/400 Network File System Support  Este manual descreve o NFS 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 neste 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, para além de conter ficheiros (de cabeçalho).
- Security — Reference  Este manual fornece informações técnicas detalhadas sobre a segurança do OS/400.
- 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.
- Cópia de Segurança e Recuperação  Este manual fornece informações gerais sobre as opções de recuperação e disponibilidade relativas ao servidor IBM iSeries.



## Bibliografia

# Índice Remissivo

## A

- API
  - ILE C/400 55
  - programa de exemplo 103
  - regras do nome de caminho 56
  - utilizar em programas C 45, 50
- arquivadores
  - sistema de ficheiros QDLS 5
- Arquivadores
  - sistema de ficheiros QDLS 77
- atributos expandidos
  - continuidade entre idiomas nacionais 23, 59
  - directrizes de nomenclatura 23
  - o que são? 22
- autoridade
  - comandos 26
  - limitações do sistema de ficheiros QNTC 84
  - limitações para o sistema de ficheiros QFileSvr.400 88
  - no programa de exemplo 103
  - tratamento em programas 58

## B

- bibliografia 111

## C

- Client Access 33
- cliente de PC
  - trabalhar com sistema de ficheiros integrado 31
- clientes de PC
  - como os sistemas de ficheiros estão representados 34
- comandos
  - lista 26
  - regras do nome de caminho 29
  - utilizar 26
- conversão
  - dados 59
  - nomes do objecto 23, 59
- conversão de caracteres 2, 23, 59
- conversão de dados 59

## D

- descrição de ficheiro aberto 57
- descriptor de ficheiros 57
- directório
  - actual 8
  - benefício 1
  - comandos 26
  - inicial 8
  - menus e ecrãs 26
  - no programa de exemplo 103
  - o que é? 6
  - sistema de ficheiros integrado 35

- directório actual 8
- directório de trabalho 8
- directório inicial 8

## E

- ecrãs
  - regras do nome de caminho 29
  - utilizar 25
- espaço de utilizador
  - utilizar no sistema de ficheiros QSYS.LIB 72
  - utilizar no sistema de ficheiros QSYS.LIB de ASP Independente 75
- esquemas de codificação 23, 59
- exemplo
  - nomes de caminho 18, 29, 56
  - programa utilizando APIs do sistema de ficheiros integrado 103
  - utilizar ligação simbólica 20

## F

- ficheiro 103
  - menus e ecrãs 26
  - modos de abertura 59
  - transferir 32
- ficheiro de base de dados
  - comparação com ficheiros de dados contínuos 3
  - copiar de/para ficheiro de dados contínuos 46
  - criar a partir de ficheiro de dados contínuos 46
- ficheiro de dados contínuos
  - benefício 1
  - comparação com ficheiro orientado para registos 3
  - copiar para/de ficheiro de base de dados 46
  - indicar a utilização no ILE C/400 55
  - no programa de exemplo 103
  - o que é? 3
  - por que utilizar? 3
  - utilizar em programas 45
- ficheiros de salvaguarda
  - utilizar no sistema de ficheiros QSYS.LIB 73
  - utilizar no sistema de ficheiros QSYS.LIB de ASP Independente 75
- FTP 32
- funções
  - ILE C/400 55
  - no programa de exemplo 103
  - regras do nome de caminho 56
  - utilizar em programas C 45, 50

## I

- ILE C/400
  - APIs alternativas 50
  - funções de ANSI 55
- indicadores 57
- Informação Impressa Relacionada 111

Interface de utilizador  
  comandos 26  
  menus e ecrãs 25  
  visualizar a partir de PCs 34  
interface do sistema de ficheiros integrado 1, 2, 5

## L

ligação  
  comandos 26  
  comparação 21  
  menus e ecrãs 26  
  no programa de exemplo 103  
  o que é? 18  
  permanente 19  
  por que utilizar? 18  
  simbólica 20  
  utilizar no sistema de ficheiros / (raiz) 65  
  utilizar no sistema de ficheiros QDLS 78  
  utilizar no sistema de ficheiros QFileSvr.400 89  
  utilizar no sistema de ficheiros QNTC 85  
  utilizar no sistema de ficheiros QOpenSys 67  
  utilizar no sistema de ficheiros QOPT 80  
  utilizar no sistema de ficheiros QSYS.LIB 73  
  utilizar no sistema de ficheiros QSYS.LIB de ASP  
    Independente 76  
ligação permanente  
  comparação com ligação simbólica 21  
  o que é? 19  
ligação simbólica  
  comparação com ligação permanente 21  
  exemplo de utilização 20  
  o que é? 20  
LU 6.2 no sistema de ficheiros QFileSvr.400 87

## M

menus  
  regas do nome de caminho 29  
  utilizar 25  
modo de acesso 58  
modo de ficheiro aberto binário 59  
modo de ficheiro aberto de texto 59  
modos de ficheiro aberto 59

## N

National Language Support 2, 23, 59  
NetServer 33  
nome de caminho

  nome de caminho absoluto 18  
  nome de caminho relativo 18  
  o que é? 17  
  regas para APIs 56  
  regas para comandos e ecrãs 29  
  utilizar no sistema de ficheiros / (raiz) 65  
  utilizar no sistema de ficheiros QDLS 78  
  utilizar no sistema de ficheiros QFileSvr.400 87  
  utilizar no sistema de ficheiros QNTC 84  
  utilizar no sistema de ficheiros QOpenSys 66  
  utilizar no sistema de ficheiros QOPT 80

nome de caminho (*continuação*)  
  utilizar no sistema de ficheiros QSYS.LIB 73  
  utilizar no sistema de ficheiros QSYS.LIB de ASP  
    Independente 76

nome de caminho absoluto 18  
nome de caminho relativo 18

nomes

  continuidade entre esquemas de codificação 23  
  continuidade entre idiomas nacionais 59  
  utilizar no sistema de ficheiros / (raiz) 65  
  utilizar no sistema de ficheiros QDLS 78  
  utilizar no sistema de ficheiros QFileSvr.400 87  
  utilizar no sistema de ficheiros QNTC 84  
  utilizar no sistema de ficheiros QOpenSys 66  
  utilizar no sistema de ficheiros QOPT 80  
  utilizar no sistema de ficheiros QSYS.LIB 73  
  utilizar no sistema de ficheiros QSYS.LIB de ASP  
    Independente 76

## O

objecto  
  migrar através de sistemas de ficheiros 34  
operações (programa de exemplo) 103

## P

página de códigos 2, 23, 59  
permissão 58  
processador de I/O do servidor de ficheiros 6  
programa de linguagem C  
  exemplo 103  
  funções de ILE C/400 55  
Protocolo de Transferência de Ficheiros 32

## Q

QFileSvr.400 5  
QOPT 5  
QSYS.LIB de ASP Independente  
  características e limitações 74

## R

Registar objectos do sistema de ficheiros integrado em  
  diário 93  
Registo em diário  
  Iniciar 42  
  Terminar 42

## S

segurança  
  comandos 26  
  limitações do sistema de ficheiros QNTC 84  
  limitações para o sistema de ficheiros  
    QFileSvr.400 88  
  tratamento em programas 58  
servidor de ficheiros 6  
servidor de ficheiros de PC 6

servidor de ficheiros do OS/400 6

sistema de ficheiros

- benefício 2
- biblioteca (QSYS.LIB)
  - características e limitações 71
  - descrição 5
- comparação 61
- interface 5
- migrar objectos 34
- o que é? 4
- óptico (QOPT)
  - características e limitações 79
- QSYS.LIB de ASP independente
  - descrição 5
- QSYS.LIB de ASP Independente
  - características e limitações 74
- raiz (/)
  - características e limitações 64
  - descrição 4
- serviços da biblioteca de documentos (QDLS)
  - descrição 5
- serviços de biblioteca de documentos (QDLS)
  - características e limitações 77
- servidor de ficheiros do OS/400 (QFileSvr.400)
  - características e limitações 86
- servidor QNTC
  - características e limitações 84
- Sistema de ficheiros da rede
  - descrição 5
- Sistema de Ficheiros da Rede
  - características e limitações 90
- sistema de ficheiros definido pelo utilizador
  - descrição 4
- sistema de ficheiros Definido pelo Utilizador
  - características e limitações 67
- sistema de ficheiros NetWare (QNetWare)
  - características e limitações 81
- Sistema de ficheiros óptico (QOPT)
  - descrição 5
- sistema de ficheiros QFileSvr.400 (QFileSvr.400)
  - descrição 5
- Sistema de ficheiros QNetWare
  - descrição 5
- sistema de ficheiros QNTC
  - descrição 5
- sistemas abertos (QOpenSys)
  - características e limitações 66
  - descrição 4
- transferir ficheiros 32

sistema de ficheiros da rede 5

- descrição 5

Sistema de Ficheiros da Rede

- características e limitações 90

sistema de ficheiros de biblioteca (QSYS.LIB)

- características e limitações 71
- descrição 5

sistema de ficheiros de serviços de biblioteca de documentos (QDLS)

- características e limitações 77

sistema de ficheiros de sistemas abertos (QOpenSys)

- características e limitações 66

sistema de ficheiros de sistemas abertos (QOpenSys)

- (*continuação*)
- descrição 4

sistema de ficheiros definido pelo utilizador 4

- descrição 4

sistema de ficheiros Definido pelo Utilizador

- características e limitações 67

sistema de ficheiros do QOPT

- descrição 5

sistema de ficheiros do servidor de ficheiros (QFileSvr.400) do OS/400

- características e limitações 86

Sistema de Ficheiros do Windows NT Server (QNTC)

- características e limitações 84

sistema de ficheiros dos serviços de biblioteca de documentos (QDLS)

- descrição 5

sistema de ficheiros hierárquico (HFS)

- utilizar APIs para o sistema de ficheiros QDLS 77
- utilizar APIs para o sistema de ficheiros QOPT 79

sistema de ficheiros integrado

- comandos
  - lista 26
  - regras do nome de caminho 29
  - utilizar 26
- interfaces de programação
  - indicadores e descritores de ficheiros 57
  - National Language Support 59
  - programa de exemplo 103
  - regras do nome de caminho 56
  - segurança 58
  - utilizar em programas C 50
- menus e ecrãs
  - regras do nome de caminho 29
  - utilizar 25
- o que é? 1
- porque deve ser utilizado? 1
- trabalhar a partir de PCs
  - como os sistemas de ficheiros estão representados 34
- trabalhar a partir de um PC
  - interacção com directórios e objectos 31

sistema de ficheiros NetWare (QNetWare)

- características e limitações 81

sistema de ficheiros óptico (QOPT)

- características e limitações 79
- descrição 5

sistema de ficheiros QDLS

- características e limitações 77
- descrição 5

sistema de ficheiros QFileSvr.400

- características e limitações 86
- descrição 5

sistema de ficheiros QFileSvr.400 (QFileSvr.400)

- descrição 5

Sistema de ficheiros QNetWare 5

- características e limitações 81
- descrição 5

sistema de ficheiros QNTC 5

- características e limitações 84
- descrição 5

- sistema de ficheiros QOpenSys
  - características e limitações 66
  - descrição 4
- sistema de ficheiros QOPT
  - características e limitações 79
- sistema de ficheiros QSYS.LIB
  - características e limitações 71
  - descrição 5
- sistema de ficheiros QSYS.LIB de ASP independente
  - descrição 5
- sistema de ficheiros raiz (/)
  - características e limitações 64
  - descrição 4
- socket 58

## **T**

- TCP/IP no sistema de ficheiros QFileSvr.400 87
- transferência através de sistemas de ficheiros 34

## **U**

- unicode 23



**IBM**