



Sistemas IBM - iSeries

Ficheiros e sistemas de ficheiros

Sistema de ficheiros integrado

Versão 5 edição 4





Sistemas IBM - iSeries

Ficheiros e sistemas de ficheiros
Sistema de ficheiros integrado

Versão 5 edição 4

Atenção

Antes de utilizar as informações contidas nesta publicação, leia as informações incluídas na secção "Avisos", na página 151.

Sétima edição (Fevereiro de 2006)

Esta edição é referente à versão 5, edição 4, modificação 0 do IBM i5/OS (número de produto 5722-SS1) e a todas as edições e modificações posteriores deste programa, até indicação em contrário fornecida através de novas edições desta publicação. Esta versão não é executada em todos os modelos de sistemas Reduced Instruction Set Computer (RISC), nem nos modelos de sistemas CISC.

© Copyright International Business Machines Corporation 1999, 2006. Todos os direitos reservados.

Índice

Sistema de ficheiros integrado	1
Novidades na versão V5R4	1
PDF imprimível	2
Descrição geral do sistema de ficheiros integrado	2
Em que consiste o sistema de ficheiros integrado	2
Por que motivo utilizar o sistema de ficheiros integrado	3
Conceitos sobre o sistema de ficheiros integrado	4
Directório	4
Ligação.	12
Nome de caminho	16
Ficheiro de dados contínuos.	18
Continuidade de nomes	19
Atributos expandidos	20
Suporte de pesquisa	21
Trabalhar com sistemas de ficheiros	27
Comparação entre sistemas de ficheiros	29
sistema de ficheiros "raiz" (/)	33
Sistema de ficheiros de sistemas abertos (QOpenSys)	36
Sistemas de ficheiros definidos pelo utilizador (UDFSs)	38
Sistema de ficheiros de biblioteca (QSYS.LIB)	45
QSYS.LIB de ASP independente	48
Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)	52
Sistema de ficheiros óptico (QOPT)	54
Sistema de ficheiros NetWare (QNetWare)	57
Sistema de ficheiros do iSeries NetClient (QNTC)	60
Sistema de ficheiros do servidor de ficheiros do i5/OS (QFileSvr.400)	65
Network File System (NFS)	69
Aceder ao sistema de ficheiros integrado	72
Aceder utilizando menus e ecrãs	72
Aceder utilizando comandos de CL	74
Aceder utilizando APIs	93
Aceder utilizando o iSeries Navigator	93
Aceder utilizando o iSeries NetServer	94
Acesso através do Protocolo de Transferência de Ficheiros	95
Aceder utilizando um PC.	95
Converter directórios *TYPE1 em *TYPE2	96
Descrição geral da conversão de *TYPE1 em *TYPE2.	97
Considerações sobre conversão	97
Registrar objectos em diário	102
Descrição geral do registo em diário.	102
Iniciar registo em diário	107
Alterar registo em diário	107
Terminar registo em diário	108
Regenerar sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador	108
Comparação entre os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG)	109
Comando Regenerar ligações de objectos (RCLLNK)	110
Recriar objectos fornecidos pelo sistema de ficheiros integrado	111
Exemplos: comando Regenerar ligações de objectos (RCLLNK)	111
Suporte de programação.	113
Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados	113
Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda	118
Executar operações utilizando APIs	118
Suporte de sockets	128
Nomenclatura e suporte internacional	129
Conversão de dados	129
Exemplo: Funções C do sistema de ficheiros integrado	130
Trabalhar com ficheiros e pastas através da utilização do iSeries Navigator	136
Dar entrada de um ficheiro.	136
Dar saída de um ficheiro	136
Criar uma pasta	136
Remover uma pasta	136
Mover ficheiros ou pastas para outro sistema de ficheiros	137
Definir permissões	138
Configurar conversão de texto do ficheiro	138
Enviar um ficheiro ou uma pasta para outro sistema	139
Alterar opções para a definição do pacote	139
Marcar uma data e uma hora para enviar o ficheiro ou a pasta.	140
Criar uma partilha de ficheiros	140
Alterar uma partilha de ficheiros	140
Criar um novo sistema de ficheiros definido pelo utilizador	141
Montar sistema de ficheiros definido pelo utilizador.	141
Desmontar sistema de ficheiros definido pelo utilizador.	141
Definir se os objectos devem ou não ser pesquisados	142
Chamada de procedimento remoto independente de transporte	143
APIs de selecção de rede	143
APIs de conversão de nomes para endereços	143
APIs XDR (eXternal Data Representation)	144
APIs de autenticação	145
APIs de RPC independente de transporte (TI-RPC)	146
Informações relacionadas com o sistema de ficheiros integrado.	147
Apêndice. Avisos.	151
Informações sobre Interfaces de Programação.	152

Marcas Comerciais	152	Termos e Condições	153
-----------------------------	-----	------------------------------	-----

Sistema de ficheiros integrado

O sistema de ficheiros integrado é uma parte do i5/OS™ que suporta a entrada/saída de sequência e gestão do armazenamento semelhante ao computador pessoal e aos sistemas operativos UNIX®, fornecendo ao mesmo tempo uma estrutura de integração de todas as informações armazenadas no servidor.

Nota: Ao utilizar os exemplos de código está implícita a aceitação dos termos do “Informações de licenças para códigos e de exclusão de responsabilidade” na página 149.

Novidades na versão V5R4

Este tópico destaca as alterações efectuadas nesta colecção de tópicos da versão V5R4.

QNTC

- | O sistema de ficheiros QNTC suporta agora TCP/IP porta 445 e tamanhos de ficheiro superiores.
- | • O sistema de ficheiros QNTC suporta TCP/IP porta 445
- | O sistema de ficheiros QNTC consegue agora contactar servidores utilizando TCP/IP porta 445. Não é necessário configurar os servidores do Windows® para NetBios através de TCP/IP.
- | • O sistema de ficheiros QNTC suporta tamanhos de ficheiros grandes
- | A partir da versão V5R4, o sistema de ficheiros QNTC passou a suportar a leitura e gravação de ficheiros até um tamanho de 1 TB (1 TB equivale a aproximadamente 1 099 511 627 776 bytes).

Comando Regenerar ligações de objectos (RCLLNK)

- | O comando Regenerar ligações de objectos (RCLLNK) identifica e repara objectos danificados nos sistemas de ficheiros “raiz” (/), QOpenSys e montados definidos pelo utilizador sem ser requerido que o sistema esteja no estado restrito. Deste modo, pode corrigir problemas nestes sistemas de ficheiros sem comprometer a produtividade. Pode ser utilizado como uma alternativa ao comando Regenerar Armazenamento (RCLSTG) em várias situações. Por exemplo, RCLLNK é ideal para identificar e corrigir problemas nas seguintes situações:
- | • Problemas isolados num único objecto.
- | • Problemas isolados num grupo de objectos.
- | • Objectos danificados que necessitem ser identificados ou eliminados.
- | • O sistema não pode estar no estado restrito durante a operação de regeneração.
- | • As áreas de memória auxiliar (ASP - Auxiliary Storage Pool) independentes têm de estar disponíveis durante a operação de regeneração.

Como visualizar as novidades ou o que foi alterado

Para ajudar a visualizar as alterações técnicas efectuadas, estas informações utilizam:

- A imagem  para marcar a localização em que as informações novas ou alteradas começam.
- A imagem  para marcar a localização em que as informações novas ou alteradas terminam.

Para obter outras informações sobre novidades ou sobre o que foi alterado nesta edição, consulte o tópico Memorando para os utilizadores.

PDF imprimível

Utilize este tópico para visualizar e imprimir um PDF com estas informações.

Para visualizar ou descarregar a versão em PDF deste documento, seleccione Sistemas de ficheiros integrados  (cerca de 1845 KB).

Guardar ficheiros PDF

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

1. No navegador, faça clique com o botão direito do rato no PDF pretendido (faça clique com o botão direito do rato na ligação acima).
2. Faça clique na opção que permite guardar o PDF localmente.
3. Navegue até o directório no qual pretende guardar o PDF.
4. Faça clique em **Guardar (Save)**.

Descarregar o Adobe Reader

- | Necessita de ter o Adobe Reader instalado no sistema para visualizar ou imprimir estes PDFs. Pode
- | descarregar uma cópia gratuita a partir do sítio da Web da Adobe
- | (www.adobe.com/products/acrobat/readstep.html) .

Descrição geral do sistema de ficheiros integrado

Obtenha informações sobre o sistema de ficheiros integrado no servidor iSeries™ e sobre como pode ser utilizado no seu servidor.

Em que consiste o sistema de ficheiros integrado

O *sistema de ficheiros integrado* é uma parte do i5/OS que suporta a entrada/saída de sequência e gestão do armazenamento semelhante ao computador pessoal e aos sistemas operativos UNIX, fornecendo ao mesmo tempo uma estrutura de integração de todas as informações armazenadas no servidor.

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

Os componentes chave do sistema de ficheiros integrado são os seguintes:

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

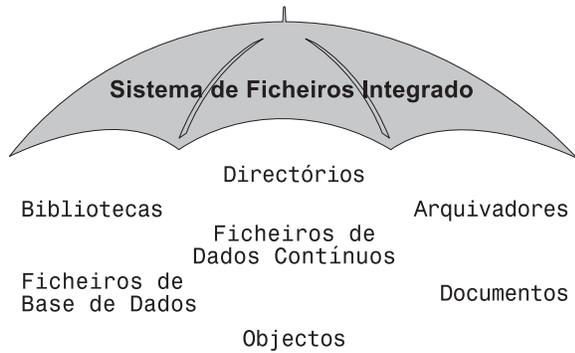


Figura 1. Uma estrutura de todas as informações armazenadas no servidor iSeries

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Por que motivo utilizar o sistema de ficheiros integrado

O sistema de ficheiros integrado melhora as capacidades já extensas da gestão de dados do i5/OS 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 executar as seguintes tarefas:

- Forneça um acesso rápido a dados do i5/OS, especialmente para aplicações como o iSeries Access, que utilizam o servidor de ficheiros do i5/OS.
- Permitir um processamento mais eficiente de tipos de ficheiros de dados contínuos como, por exemplo, imagens, áudio e vídeo.
- Forneça uma base de sistema de ficheiros e uma base de directórios para suporte de normas de sistema aberto com base no sistema operativo UNIX, como, por exemplo, POSIX (Portable Operating System Interface for Computer Environments) e X/Open Portability Guide (XPG). Esta estrutura de ficheiros e esta estrutura de directórios também fornecem um ambiente familiar para os utilizadores de sistemas operativos de PC como, por exemplo, os sistemas operativos DOS (Disk Operating System) e Windows.
- Permitir o suporte de ficheiros com capacidades únicas (tais como ficheiros de base de dados orientados para registos, ficheiros de dados contínuos com base em sistemas operativos UNIX e servidores de ficheiros) sejam processados como sistemas de ficheiros separados, sendo, no entanto, permitida a respectiva gestão através de uma interface comum.

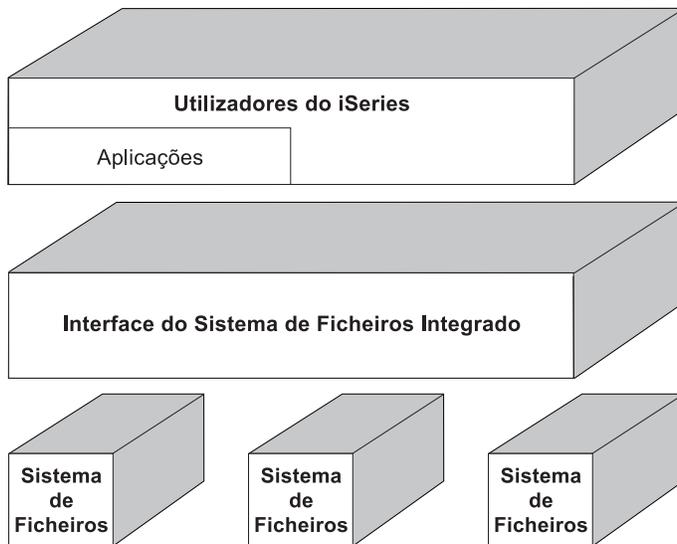


Figura 2. Interface comum para sistemas de ficheiros separados

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

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Conceitos sobre o sistema de ficheiros integrado

Este tópico apresenta os conceitos básicos sobre o sistema de ficheiros integrado, como, por exemplo, directório, ligação, nome do caminho, ficheiro de dados contínuos, continuidade de nomes, atributos expandidos e suporte de pesquisa.

Directório

Um *directório* é um objecto especial 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 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. Estas ramificações de directórios apresentam ramificações subordinadas denominadas 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.

Uma determinada ramificação de directórios, juntamente com as respectivas ramificações subordinadas (subdirectórios) e todos os objectos ligados a estas ramificações, designa-se por *subárvore*. Cada sistema de ficheiros é uma subárvore principal na estrutura de directórios do sistema de ficheiros integrado. Nas subárvores dos sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente, uma biblioteca é processada da mesma forma que um subdirectório. Os objectos numa biblioteca são processados como objectos num subdirectório. Uma vez que os ficheiros de base de dados contêm objectos (membros de ficheiros de base de dados), são processados como subdirectórios e não como objectos. No sistema de ficheiros de serviços da biblioteca de documentos (subárvore QDLS), as pastas são processadas como subdirectórios e os documentos das pastas são processados como objectos num subdirectório.

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

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

Os sistemas de ficheiros e objectos são ramificações da árvore de directórios do sistema de ficheiros integrado. Consulte a figura que se segue para obter um exemplo de uma árvore de directórios do sistema de ficheiros integrado.

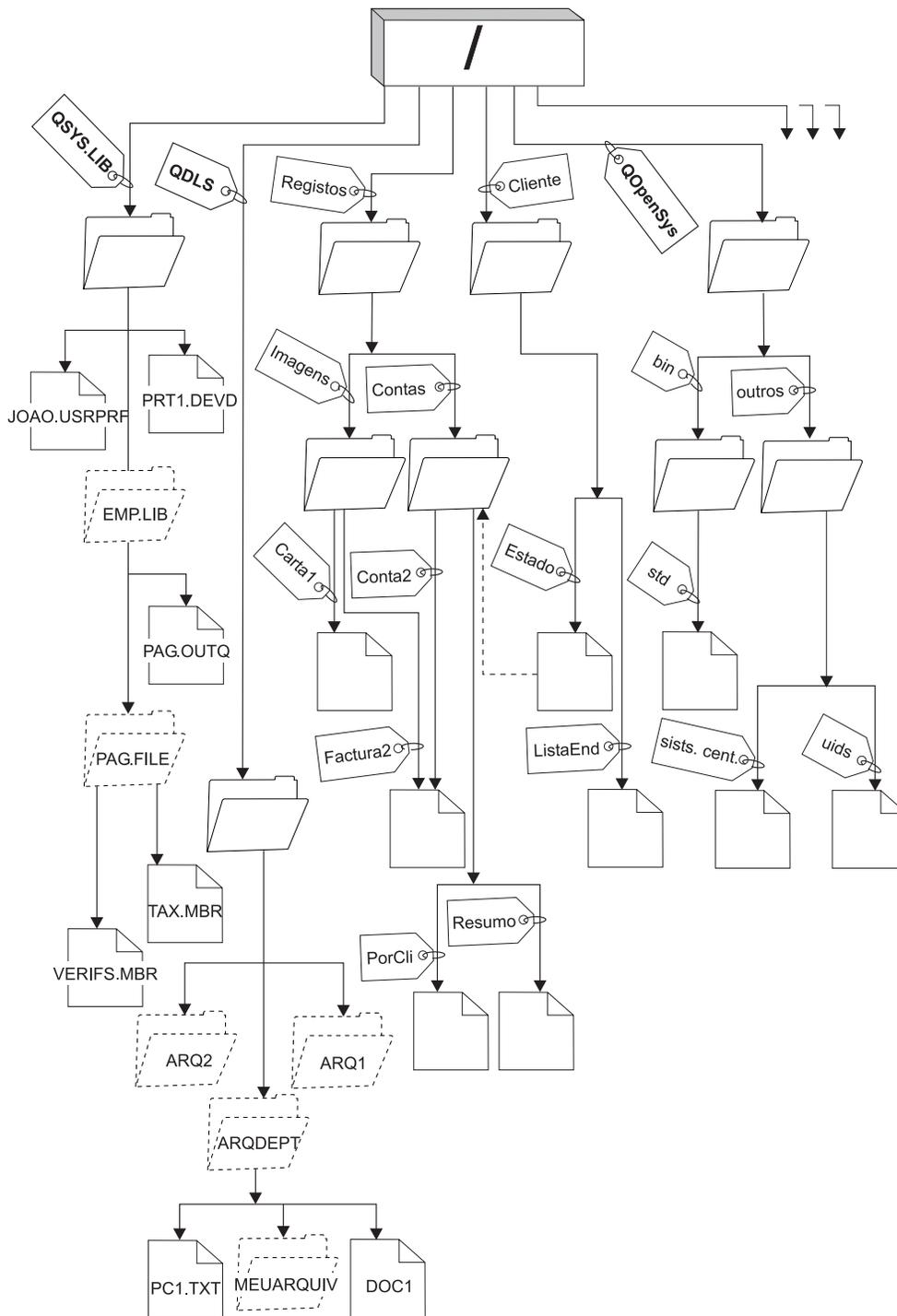


Figura 3. Amostra da árvore de directórios do sistema de ficheiros integrado

Directório actual

O directório actual é semelhante à ideia de biblioteca actual. Também é chamado *directório de trabalho actual* ou apenas *directório de trabalho*.

O *directório actual* é o primeiro directório no qual o sistema operativo efectua a procura de programas e ficheiros e armazena os ficheiros temporários e a saída. 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.

Directório inicial

O *directório inicial* é utilizado como o directório actual quando inicia sessão no sistema. O nome do directório inicial é especificado no seu perfil do utilizador.

Quando o trabalho é iniciado, o sistema procura no seu perfil de utilizador o nome do directório inicial. Se não existir um directório com esse nome no sistema, o directório inicial é alterado para o directório "raiz" (/).

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

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

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

Como valor predefinido, 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 estiver a ser utilizado por um módulo para o directório inicial do perfil do utilizador actual desse módulo (ou para o directório "raiz" (/), se esse directório inicial não existir). Os módulos secundários herdarão sempre o directório inicial do módulo que o criou. Tenha em atenção que o directório actual do processo não é alterado ao utilizar `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.

Conceitos relacionados

"Directórios fornecidos"

Ao reiniciar o sistema, o sistema de ficheiros integrado cria estes directórios, caso ainda não existam.

Informações relacionadas

Comando Alterar directório actual (`CHGCURDIR`)

`chdir()`

`fchdir()`

Interfaces de Programação de Aplicações (APIs)

Directórios fornecidos

Ao reiniciar o sistema, o sistema de ficheiros integrado cria estes directórios, caso ainda não existam.

Nota: Não substitua os directórios criados pelo sistema que se seguem por ligações simbólicas a outros objectos. Por exemplo, não substitua /home por uma ligação simbólica a um directório numa ASP independente. Caso contrário, poderão ocorrer problemas na ASP independente, bem como ao criar novos perfis de utilizador.

/tmp O directório /tmp fornece às aplicações um local onde armazenar os objectos temporários. Trata-se de um subdirectório do directório "raiz" (/), pelo que o respectivo nome de caminho é /tmp.

Uma vez colocado por determinada aplicação no directório /tmp, o objecto permanece nesse directório até ser removido pelo utilizador ou por uma aplicação. O sistema não remove automaticamente objectos do directório /tmp nem efectua qualquer outro processamento especial relativamente a objectos no directório /tmp.

Pode utilizar os comandos e os ecrãs do utilizador que suportem o sistema de ficheiros integrado para gerir o directório /tmp e respectivos objectos. Por exemplo, pode utilizar o ecrã Trabalhar com Ligações de Objecto ou o comando WRKLNK para copiar, remover ou mudar o nome do directório /tmp ou de objectos existentes nesse 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 programas de aplicações (API - Application Program Interfaces) que suportem o sistema de ficheiros integrado para gerir o directório /tmp e respectivos objectos. Por exemplo, o programa de aplicação pode remover um objecto no directório /tmp utilizando a API unlink().

Se o /tmp for removido, será automaticamente criado de novo durante o próximo reinício do sistema.

O directório /tmp pode ter o atributo mudança de nome e remoção de ligação restritas (restricted rename and unlink) definido com Sim (Yes) tendo em vista finalidades de segurança e compatibilidade de sistema operativo.

Nota: O atributo mudança de nome e remoção de ligação restritas (restricted rename and unlink) é equivalente ao bit de modo S_ISVTX para um directório.

Se o atributo mudança de nome e remoção de ligação restritas (restricted rename and unlink) estiver definido como Sim (Yes), não pode mudar o nome nem remover as ligações de objectos no directório /tmp, a menos que uma das seguintes condições seja verdadeira:

- É o proprietário do objecto.
- É o proprietário do directório.
- Tem autoridade especial para todos os objectos (*ALLOBJ).

Se o atributo estiver definido como Sim (Yes) e não tiver as autoridades apropriadas, visualizará o número de erro 3027 (EPERM) ou a mensagem MSGCPFA0B1 (Operação solicitada não permitida. Problema de acesso (Requested operation not allowed. Access problem)) para falhas de mudança de nome ou de remoção de ligação ao utilizar os seguintes comandos e APIs:

- Comando Remover ligação (RMVLNK, DEL e ERASE)
- Comando Remover directório (RMVDIR, RD e RMDIR)
- Comando Mudar o nome do objecto (RNM e REN)
- Comando Mover objecto (MOV e MOVE)
- API Mudar o nome do ficheiro ou do directório (rename())
- API Mudar o nome do ficheiro ou do directório, manter "novo" se existir (Qp0IRenameKeep())
- API Mudar o nome do ficheiro ou do directório, remover ligação de "novo" se existir (Qp0IRenameUnlink())
- API Remover directório (rmdir())
- API Remover ligação para o ficheiro (unlink())

O atributo mudança de nome e remoção de ligação restritas (restricted rename and unlink) e o bit de modo S_ISVTX podem ser modificados utilizando o comando Alterar atributo (CHGATR) ou as APIs Definir atributos (Qp0ISetAttr()) ou Alterar autorizações de ficheiros (chmod), se for o proprietário do objecto ou se tiver a autoridade especial para todos os objectos (*ALLOBJ). No

entanto, se o atributo for alterado para Não (No), perderá as vantagens a nível de compatibilidade e segurança do sistema operativo fornecidas pela definição Sim (Yes).

Quando o directório /tmp é criado durante um reinício do sistema, o atributo é definido como Sim (Yes). Se o directório /tmp já existir durante um reinício do sistema, o atributo não é alterado.

/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.

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

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

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

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

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

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

/QOpenSys/QIBM
O directório /QOpenSys/QIBM é o directório do sistema de ficheiros QOpenSys.

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

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

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

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

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

/dev/xti
O directório /dev/xti contém os controladores de dispositivos UDP e TCP.

Conceitos relacionados

“Directório inicial” na página 7

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.

Referências relacionadas

“Dispositivos UDP e TCP no sistema de ficheiros “raiz” (/)” na página 35

O sistema de ficheiros “raiz” (/) no directório /dev/xti suportará agora dois controladores de dispositivos denominados udp e tcp.

“Sistema de ficheiros de sistemas abertos (QOpenSys)” na página 36

O sistema de ficheiros QOpenSys é compatível com as normas do sistema aberto com base em UNIX, como POSIX e X/Open Portability Guide (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.

“Recriar objectos fornecidos pelo sistema de ficheiros integrado” na página 111

Esta tabela mostra os objectos fornecidos pelo sistema de ficheiros integrado recriados pelo comando Regenerar ligações de objectos (RCLLNK), caso não existam. Estes objectos são criados geralmente durante o carregamento do programa inicial (IPL - Initial Program Load). Se necessário, também pode recriar alguns destes objectos utilizando o comando Regenerar armazenamento (RCLSTG).

Informações relacionadas

Comando WRKLNK

Directórios *TYPE2

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

Nota: O conceito dos ficheiros de dados contínuos *TYPE1 e *TYPE2 é diferente do conceito de formatos de directório *TYPE1 e *TYPE2. Não estão relacionados entre si.

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
- Em muitos casos, menos espaço de memória auxiliar

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:

- 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).
- Um objecto num directório *TYPE2 pode ter até um milhão de ligações, em comparação com as 32 767 ligações nos directórios *TYPE1. Ou seja, pode ter até 1 milhão de caminhos directos para um ficheiro de dados contínuos e um directório *TYPE2 pode conter até 999 998 subdirectórios.
- Através do iSeries Navigator, quando abre um directório com o formato *TYPE2, a lista de entradas é automaticamente ordenada em sequência binária.
- Algumas funções novas, tais como o suporte de pesquisa do sistema de ficheiros integrado, só estão disponíveis para os objectos existentes nos directórios *TYPE2.

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

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

- Os novos servidores iSeries que sejam pré-instalados com o OS/400® V5R2 ou o i5/OS V5R3 ou posterior têm directórios *TYPE2. Não é necessária a conversão para sistemas de ficheiros "raiz" (/), QOpenSys e UDFSs nas ASPs 1-32.
- Uma instalação de raiz do OS/400 V5R2 ou do i5/OS V5R3 ou posterior num servidor iSeries tem directórios *TYPE2. Não é necessária a conversão para sistemas de ficheiros "raiz" (/), QOpenSys e UDFSs nas ASPs 1-32.
- É utilizado o utilitário de conversão da versão V5R1 ou V5R2 para converter os sistemas de ficheiros.
- Se os UDFSs existentes numa ASP independente ainda não tiverem sido convertidos para o formato *TYPE2, tal verifica-se quando a ASP independente for activada pela primeira vez num sistema com o OS/400 V5R2 ou o i5/OS V5R3 ou posterior instalado.
- Todos os outros sistemas de ficheiros suportados, excepto os UDFSs em ASPs independentes que ainda estejam a utilizar directórios *TYPE1, são convertidos automaticamente pelo sistema. Esta conversão é iniciada após a instalação do i5/OS V5R3M0 ou de versões posteriores. Não deverá afectar significativamente a actividade do sistema.

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

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

Nota: Os directórios *TYPE2 são suportados no OS/400 V5R2 ou no i5/OS V5R3 ou posterior, mas existem algumas diferenças em relação ao suporte de directório *TYPE2 normal.

Utilizar directórios *TYPE2 no OS/400 V5R1 ou V5R2

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 no OS/400 V5R1, V5R2 e posterior.

O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original. Os directórios *TYPE2 têm uma estrutura interna diferente dos directórios *TYPE1 e facultam um aumento do rendimento e fiabilidade.

Se tiver o OS/400 V5R1 ou V5R2, pode converter os directórios para o formato de directório *TYPE2 através do utilitário de conversão adequado. Pouco depois do i5/OS V5R3M0 ou de uma edição posterior ser instalada, a conversão para directórios *TYPE2 será automaticamente iniciada em qualquer dos sistemas de ficheiros que ainda não tenham sido convertidos, tendo em vista o suporte de directórios *TYPE2. Deste modo, convém efectuar a conversão para o formato de directório *TYPE2 antes de instalar uma versão posterior, de forma a evitar esta conversão automática.

O suporte de directórios *TYPE2 no OS/400 V5R2 está disponível no V5R2 iSeries Information Center através do comando Converter Directório (CVTDIR).

O suporte de directórios *TYPE2 no OS/400 V5R1 está disponível através de correcções temporárias de programas (PTFs - Program Temporary Fixes). O utilitário de conversão é ligeiramente diferente do utilitário da versão OS/400 V5R2. Consulte o APAR informativo II13161 para obter a documentação completa sobre os directórios *TYPE2 na V5R1. Utilize um dos seguintes métodos para ter acesso ao APAR:

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

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

- Consulte www.ibm.com/eserver/iseres/support/supporthome.nsf/document/10000045  para visualizar o APAR informativo. Consulte **Resolução de problemas (Problem Solving) → Bases de dados técnicas (Technical Databases) → Pedido para correcção de erros em programas IBM (APARs) [Authorized Program Analysis Reports (APARs)] → APARs V5R1 (V5R1 APARs) → Número de APAR II13161 (APAR number II13161).**

Conceitos relacionados

“Continuidade de nomes” na página 19

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

“Descrição geral da conversão de *TYPE1 em *TYPE2” na página 97

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

Referências relacionadas

“Converter directórios *TYPE1 em *TYPE2” na página 96

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

“Nomes de caminhos no sistema de ficheiros “raiz” (/)” na página 34

Os nomes de caminhos têm um formato específico no sistema de ficheiros “raiz” (/).

“Nomes de caminhos num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado” na página 40

Um ficheiro especial de blocos (*BLKSF) representa um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) quando é necessário manipular todo o UDFS e todos os objectos nele contidos.

Informações relacionadas

Comando Regenerar armazenamento (RCLSTG)

Comando Converter directório (CVTDIR)

Ligação

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

Para os utilizadores dos sistemas de ficheiros baseados em directórios, é conveniente pensar num objecto, por exemplo, um ficheiro, como algo que tem um nome que o identifica perante o servidor. De facto, é o caminho do directório para o objecto que o identifica. Por vezes, pode aceder a um objecto indicando apenas o “nome” do objecto. Pode fazer isto apenas porque o sistema está concebido de forma a assumir a parte do directório do caminho, em determinadas condições. A ideia de uma ligação tira partido da realidade de que é o caminho do directório que identifica o objecto. O nome é atribuído à ligação e não ao objecto.

Uma vez familiarizado com a ideia de que é à ligação que é atribuído o nome e não ao objecto, começa a ver possibilidades que anteriormente não eram tão evidentes. Podem existir múltiplas ligações ao mesmo objecto. Por exemplo, dois utilizadores podem partilhar um ficheiro, tendo uma ligação ao mesmo a partir do directório inicial de cada utilizador (consulte “Directório inicial” na página 7). Determinados tipos de ligações podem cruzar sistemas de ficheiros e podem existir sem haver um objecto.

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

Tabela 1. Comparação entre caminho directo e ligação simbólica

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

Referências relacionadas

“Ligações no sistema de ficheiros “raiz” (/)” na página 34

São permitidas múltiplas ligações permanentes para o mesmo objecto no sistema de ficheiros “raiz” (/). As ligações simbólicas são completamente suportadas.

“Ligações no sistema de ficheiros QOpenSys” na página 37

São permitidos vários caminhos directos para o mesmo objecto no sistema de ficheiros QOpenSys. As ligações simbólicas são completamente suportadas.

“Ligações num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado” na página 41

Um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) permite vários caminhos directos para o mesmo objecto e suporta totalmente as ligações simbólicas.

“Ligações no sistema de ficheiros QSYS.LIB” na página 47

Não é possível criar nem armazenar ligações simbólicas no sistema de ficheiros QSYS.LIB.

“Ligações no sistema de ficheiros QSYS.LIB de ASP independente” na página 51

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

“Ligações no sistema de ficheiros QDLS” na página 53

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

“Ligações no sistema de ficheiros QOPT” na página 56

O sistema de ficheiros QOPT suporta apenas uma ligação a um objecto. Não é possível criar nem armazenar ligações simbólicas em QOPT.

“Ligações no sistema de ficheiros QNTC” na página 61

O sistema de ficheiros QNTC suporta apenas uma ligação a um objecto. Não pode criar ou armazenar ligações simbólicas no QNTC.

“Ligações no sistema de ficheiros QFileSvr.400” na página 67

O sistema de ficheiros QFileSvr.400 suporta apenas uma ligação a um objecto.

“Ligações no Network File System” na página 70

Em geral, são permitidas múltiplas ligações permanentes ao mesmo objecto no Network File System.

“Sugestões: Ligação simbólica” na página 100

Ligações simbólicas são objectos existentes no sistema de ficheiros integrado que contêm um caminho para outro objecto.

“Ligações no sistema de ficheiros QNetWare” na página 59

O sistema de ficheiros QNetWare suporta apenas uma ligação a um objecto. Não é possível criar nem armazenar ligações simbólicas no QNetWare.

Ligação permanente

Um *caminho directo*, por vezes, denominado apenas ligação, não pode existir se não estiver ligado a um objecto real.

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

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

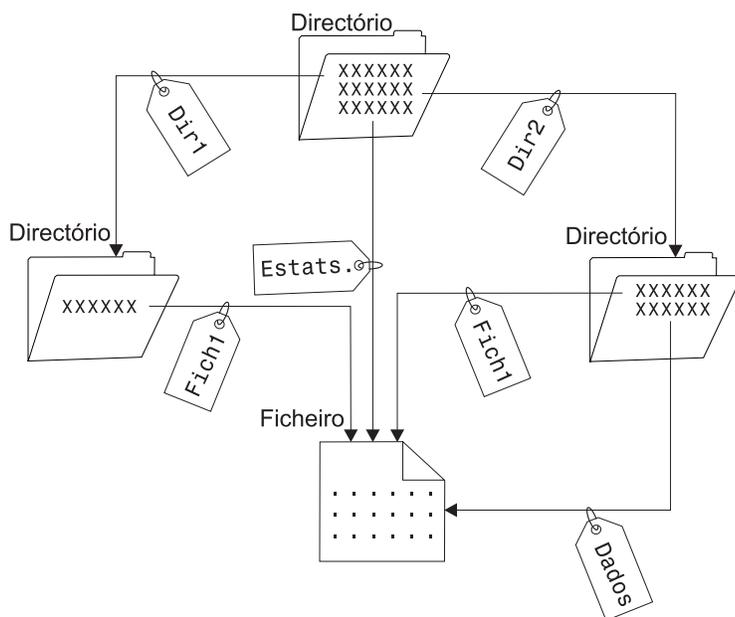


Figura 4. Uma entrada de directório define cada caminho directo

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

O conceito de caminho directo também pode ser aplicado aos sistemas de ficheiros QSYS.LIB ou QSYS.LIB de ASP independente e aos serviços da biblioteca de documentos (QDLS), mas com uma

restrição. De facto, uma biblioteca tem uma ligação permanente a cada objecto na biblioteca. Da mesma forma, uma pasta tem uma ligação permanente a cada documento existente na pasta. No entanto, não são permitidos vários caminhos directos para o *mesmo objecto* em QSYS.LIB, QSYS.LIB de ASP independente ou QDLS.

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

Conceitos relacionados

“Considerações sobre múltiplas ligações permanentes e o registo em diário” na página 106

Se tiver múltiplas ligações permanentes a um objecto do sistema de ficheiros integrado registado em diário, todas as ligações deverão ser guardadas e restauradas ao mesmo tempo para que a ligação seja mantida, bem como, as informações de registo em diário associadas.

Ligação simbólica

Uma *ligação simbólica*, também conhecida como ligação não permanente, é um nome de caminho contido num ficheiro.

Quando o sistema encontra uma ligação simbólica, segue o nome de caminho fornecido pela ligação simbólica e, em seguida, continua em qualquer caminho restante que segue a ligação simbólica. Se o nome de caminho começar por /, o sistema volta ao directório / (“raiz”) e começa a seguir o caminho a partir desse ponto. Se o nome de caminho não começar por /, o sistema regressa ao directório imediatamente anterior e segue o nome de caminho na ligação simbólica que começa nesse directório.

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

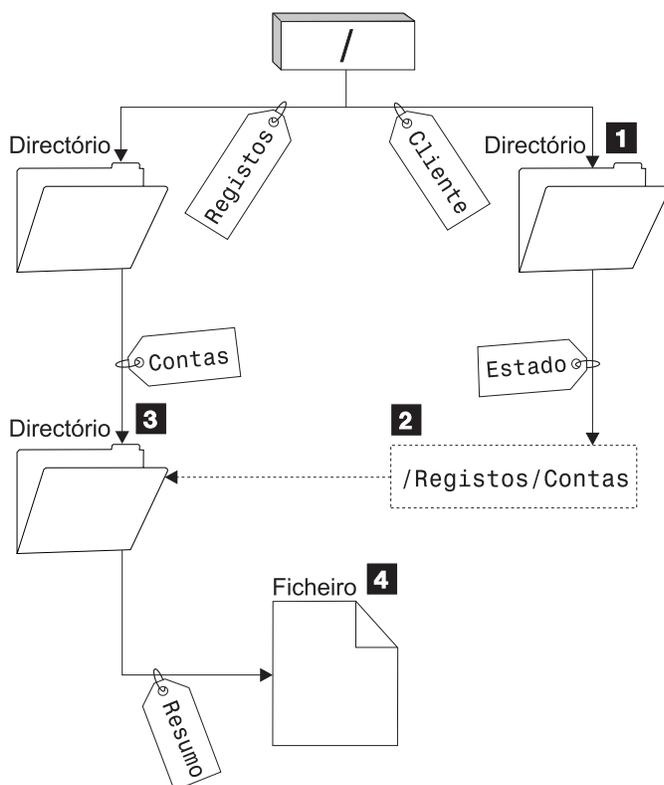


Figura 5. Exemplo de utilização de uma ligação simbólica

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

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

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

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

- Aceder a um membro do 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.

Nome de caminho

Um *nome de caminho*) indica ao servidor como localizar um objecto.

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

directório1/directório2/ficheiro

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

Há duas formas de indicar um nome de caminho:

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

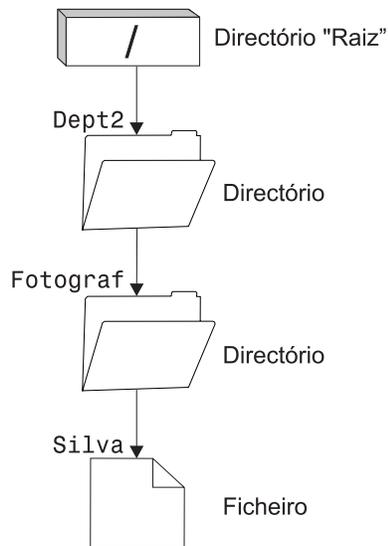


Figura 6. Componentes de um nome de caminho

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

`/Dept2/Fotograf/Silva`

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

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

`Fotograf/Silva`

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

Referências relacionadas

“Nomes de caminhos no sistema de ficheiros “raiz” (/)” na página 34

Os nomes de caminhos têm um formato específico no sistema de ficheiros “raiz” (/).

“Nomes de caminhos no sistema de ficheiros QOpenSys” na página 37

Os nomes de caminhos têm um formato específico no sistema de ficheiros QOpenSys.

“Nomes de caminhos no sistema de ficheiros QSYS.LIB de ASP independente” na página 50

Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto.

“Nomes de caminhos no sistema de ficheiros QNTC” na página 61

| O caminho é constituído pelo nome do sistema de ficheiros, pelo nome do servidor, pelo nome da
| partilha, pelos nomes de directórios e subdirectórios e pelo nome do objecto.

“Nomes de caminhos no sistema de ficheiros QFileSvr.400” na página 65

Os nomes de caminhos têm um formato específico no sistema de ficheiros QFileSvr.400.

“Nomes de caminhos no sistema de ficheiros QOPT” na página 55

O nome de caminho tem de começar por uma barra (/). O caminho é constituído pelo nome do sistema de ficheiros, pelo nome do volume, pelos nomes de directórios e subdirectórios e pelo nome do ficheiro.

“Regras de nomes de caminhos para APIs” na página 126

Quando utilizar uma API de ILE C/400® ou do sistema de ficheiros integrado para operar um objecto, o objecto é identificado fornecendo o respectivo caminho do directório. Segue-se um resumo das regras a ter em consideração ao especificar nomes de caminhos nas APIs.

“Regras de nomes de caminhos para comandos de CL e ecrãs” na página 77

Ao utilizar um comando ou ecrã do sistema de ficheiros integrado para operar um objecto, este é identificado fornecendo o respectivo nome de caminho.

“Nomes de caminhos no sistema de ficheiros QSYS.LIB” na página 47

Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto.

“Nomes de caminhos no sistema de ficheiros QDLS” na página 53

Cada componente do nome do caminho pode ser constituído apenas por um nome.

“Nomes de caminhos num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado” na página 40

Um ficheiro especial de blocos (*BLKSF) representa um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) quando é necessário manipular todo o UDFS e todos os objectos nele contidos.

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 no formato de ficheiros de dados contínuos. Os documentos armazenados nas pastas do servidor são ficheiros de dados contínuos. Outros exemplos de ficheiros de dados contínuos são os ficheiros de PC e os ficheiros dos sistemas UNIX. Um ficheiro de dados contínuos do sistema de ficheiros integrado é um objecto do sistema que tem um tipo de objecto *STMF.

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



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

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

uma imagem do cliente, compostas por uma cadeia contínua de bits representando variações de cor. Os ficheiros de dados contínuos são particularmente adequados para armazenamento de cadeias de dados, tais como o texto de um documento, imagens, áudio e vídeo.

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

Nota: O conceito dos ficheiros de dados contínuos *TYPE1 e *TYPE2 é diferente do conceito de formatos de directório *TYPE1 e *TYPE2. Não estão relacionados entre si.

Ficheiros de dados contínuos *TYPE1

Um ficheiro de dados contínuos *TYPE1 tem o mesmo formato dos ficheiros de dados contínuos criados em edições anteriores ao OS/400 V4R4.

- | O ficheiro de dados contínuos *TYPE1 tem um tamanho mínimo de 4096 bytes. Os ficheiros de dados
- | contínuos *TYPE1 têm um tamanho de objecto máximo de aproximadamente 128 GB (1 GB equivale
- | aproximadamente a 1 073 741 824 bytes).

Ficheiros de dados contínuos *TYPE2

Um ficheiro de dados contínuos *TYPE2 tem um acesso a ficheiros de alto desempenho.

Os ficheiros de dados contínuos *TYPE2 têm um tamanho de objecto máximo de aproximadamente 1 TB (1 TB equivale a aproximadamente 1 099 511 627 776 bytes) nos sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador. Caso contrário, o tamanho máximo é de aproximadamente 256 GB. Tem igualmente capacidade de definição da memória, bem como de especificação de um atributo para otimizar a atribuição de memória principal. Todos os ficheiros criados com o OS/400 V4R4 e sistemas mais recentes são ficheiros de dados contínuos *TYPE2, excepto se forem criados num sistema de ficheiros definido pelo utilizador, com um formato de ficheiro *TYPE1 especificado.

Nota: Os ficheiros com um tamanho superior a 256 GB não podem ser guardados nem restaurados em sistemas anteriores ao i5/OS V5R3.

Referências relacionadas

"Sistemas de ficheiros definidos pelo utilizador (UDFSs)" na página 38

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System) residem no conjunto de memória auxiliar (ASP - Auxiliary Storage Pool) ou no conjunto de memória auxiliar (ASP) independente à sua escolha. Pode criar e gerir estes sistemas de ficheiros.

"Copiar dados entre ficheiros de dados contínuos e ficheiros de base de dados" na página 113

Se estiver familiarizado com as operações nos ficheiros de base de dados utilizando serviços orientados para os registos, tais como especificações de descrição de dados (DDS - Data Description Specification), pode encontrar algumas diferenças fundamentais na utilização de ficheiros de dados contínuos.

Continuidade de nomes

Ao utilizar os sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

Isto também se aplica à utilização destes sistemas de ficheiros no servidor 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. O UCS2 Nível 1 e o UTF-16 são subconjuntos da norma ISO 10646. Quando o nome é utilizado, o sistema converte o formato armazenado dos caracteres na representação dos caracteres adequada na página de códigos a ser

utilizada. Os nomes de atributos expandidos associados a cada objecto também são tratados do mesmo modo.

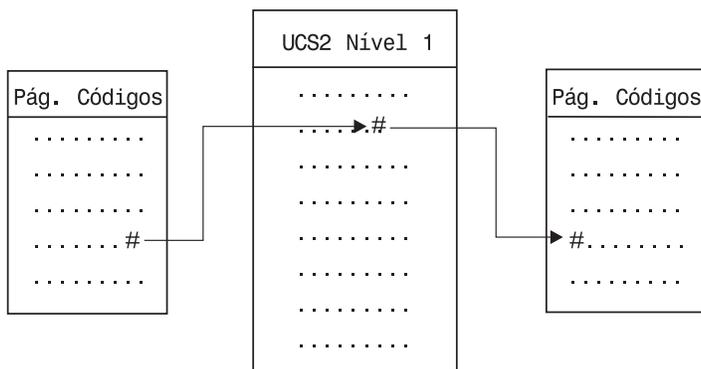


Figura 8. Manter caracteres inalterados independentemente dos esquemas de codificação

Este suporte facilita a interacção com um servidor a partir de dispositivos que utilizam diferentes páginas de códigos. Por exemplo, os utilizadores de PC podem ter acesso a um ficheiro do servidor iSeries utilizando o mesmo nome de ficheiro, mesmo que os respectivos PCs não tenham a mesma página de códigos do 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.

Conceitos relacionados

“Directórios *TYPE2” na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

Referências relacionadas

“Nomes de caminhos no sistema de ficheiros “raiz” (/)” na página 34

Os nomes de caminhos têm um formato específico no sistema de ficheiros “raiz” (/).

“Nomes de caminhos num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado” na página 40

Um ficheiro especial de blocos (*BLKSF) representa um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) quando é necessário manipular todo o UDFS e todos os objectos nele contidos.

“Nomes de caminhos no sistema de ficheiros QFileSvr.400” na página 65

Os nomes de caminhos têm um formato específico no sistema de ficheiros QFileSvr.400.

“Nomenclatura e suporte internacional” na página 129

O suporte dos sistemas de ficheiros “raiz” (/) e QOpenSys garante que os caracteres nos nomes de objectos permaneçam constantes nos esquemas de codificação utilizados para diferentes idiomas nacionais e dispositivos.

Atributos expandidos

Um atributo expandido corresponde a informações associadas a um objecto que fornecem detalhes adicionais sobre o objecto. O atributo expandido é constituído por um nome, que é utilizado para fazer referência ao mesmo, e por um valor. O valor pode ser texto, dados binários ou outro tipo de dados.

Os atributos expandidos de um objecto existem apenas enquanto o objecto existir.

Os atributos expandidos são apresentados sob diversas formas e podem ser utilizados para conter várias informações. Poderá ser necessário reconhecer os três atributos expandidos que se seguem, 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 atributos expandidos associados ao objecto.

Um ponto (.) como primeiro carácter do nome significa que o atributo expandido é um atributo expandido do sistema padrão (SEA - System Extended Attribute), que está reservado para utilização pelo sistema.

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

| O comando Trabalhar com ligações de objectos (WRKLNK) e o comando Apresentar ligações de objectos (DSPLNK) podem ser utilizados para apresentar o atributo expandido .SUBJECT de um objecto. Não existe qualquer outro suporte do sistema de ficheiros integrado através do qual as aplicações ou os utilizadores possam aceder e alterar os atributos expandidos. As únicas excepções a esta regra são os comandos de CL Apresentar UDFS (DSPUDFS) e Apresentar informações sobre sistemas de ficheiros instalados (DSPMFSINF), que apresentam os atributos expandidos aos utilizadores.

No entanto, os atributos expandidos associados a alguns objectos em QDLS podem ser alterados através de interfaces fornecidas pelo sistema de ficheiros hierárquico (HFS - Hierarchical File System).

Se um PC de 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 atributos expandidos de qualquer objecto de ficheiro. Os utilizadores do OS/2 também podem alterar os atributos expandidos de um objecto na área de trabalho, utilizando o bloco de notas de definições; isto é, seleccionando Definições (Settings) no menu emergente associado ao objecto.

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

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

NomeEmpresaNomeProduto.Nome_Atributo

Suporte de pesquisa

Com o iSeries, pode pesquisar objectos do sistema de ficheiros integrado.

Este suporte proporciona maior flexibilidade aos utilizadores do iSeries, permitindo a execução de pesquisas de vários itens; os utilizadores decidem quando pretendem que seja efectuada a pesquisa e quais as acções a executar, com base nos resultados das respectivas pesquisas.

Os dois pontos de saída relacionados com este suporte são:

- QIBM_QP0L_SCAN_OPEN - Pesquisa do Sistema de Ficheiros Integrado num Programa de Saída Aberto

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

- QIBM_QP0L_SCAN_CLOSE - Pesquisa do Sistema de Ficheiros Integrado num Programa de Saída Fechado

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

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

Tarefas relacionadas

“Definir se os objectos devem ou não ser pesquisados” na página 142

Execute os passos seguintes para definir se um objecto deve ou não ser pesquisado.

Referências relacionadas

“Pesquisa no sistema de ficheiros integrado” na página 101

Os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros de utilizador UDFS ASP só serão pesquisados utilizando os pontos de saída relacionados com a pesquisa, quando os sistemas de ficheiros tiverem sido totalmente convertidos para o formato de directório *TYPE2.

Informações relacionadas

QIBM_QP0L_SCAN_OPEN

QIBM_QP0L_SCAN_CLOSE

Exemplos: Pesquisar vírus e ficheiros a serem abertos

Estes exemplos mostram qual o programa de saída que pode pesquisar.

- Vírus

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

- Chamadas para saber quando foi aberto um ficheiro

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

A pesquisa poderá ocorrer em dois períodos de tempo diferentes, consoante a definição dos valores do sistema e do modo como é estabelecido o ambiente de pesquisa. A lista que se segue descreve tipos de pesquisa diferentes consoante a altura em que ocorrem.

1. Pesquisa em tempo de execução

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

Exemplo de verificação da existência de vírus em tempo de execução

Pretende aceder a um ficheiro do sistema de ficheiros integrado a partir do PC. Ao abrir o ficheiro a partir do PC, é efectuada a pesquisa no mesmo. Este procedimento verifica-se uma vez que o programa de saída está registado e o valor de sistema QSCANFS está definido para pesquisar ficheiros nos sistemas de ficheiros "raiz" (/), QOpenSys e UDFS. A pesquisa mostra que foi encontrado um vírus e o programa de saída anti-vírus procede à reparação do problema. Após o

programa de saída reparar o ficheiro, este deixa de estar contaminado. Deste modo, o acesso a partir do PC não é contaminado, impedindo que a contaminação seja disseminada.

E se, em vez de pesquisar a existência de vírus nesse acesso, optasse por não efectuar uma pesquisa em tempo de execução? Em seguida, após aceder ao ficheiro contaminado a partir do PC, o vírus poderá ser transferido para o PC. Ao efectuar uma pesquisa em tempo de execução, o vírus pode ser detectado antes de se disseminar pelo PC.

O principal problema deste método é a necessidade de tempo de recurso para efectuar as pesquisas. Os utilizadores que tentem aceder a um ficheiro têm de aguardar até à conclusão da pesquisa, para ser possível utilizar o ficheiro. O sistema permite assegurar que a pesquisa é efectuada apenas quando necessário e não em cada acesso.

2. Pesquisa activada manualmente ou em massa

Pode utilizar esta opção se pretende pesquisar vários itens em simultâneo. Neste caso, pode definir a pesquisa para que ocorra quando o servidor estiver inactivo, como, por exemplo, no fim-de-semana. Deste modo, haverá um menor impacto no acesso aos ficheiros durante as actividades diárias normais. A pesquisa é efectuada offline. Portanto, poderá reduzir o tempo de sistema da pesquisa em tempo de execução para ficheiros que não sejam alterados após a pesquisa em massa estar concluída. Esta situação verifica-se uma vez que não são requeridas novas pesquisas quando é efectuado novamente o acesso a esse tipo de ficheiros.

Conceitos relacionados

“Valores do sistema relacionados”

Existem dois valores de sistema relacionados com este suporte de pesquisa. Poderá utilizar estes dois valores de sistema para definir o ambiente de pesquisa que pretende para o servidor.

Informações relacionadas

QIBM_QP0L_SCAN_OPEN

QIBM_QP0L_SCAN_CLOSE

Valores do sistema relacionados

Existem dois valores de sistema relacionados com este suporte de pesquisa. Poderá utilizar estes dois valores de sistema para definir o ambiente de pesquisa que pretende para o servidor.

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

O nome e as descrições destes valores de sistema são apresentados em seguida:

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

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

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

2. Controlo de pesquisa (QSCANFSCTL)

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

- Pesquisar acessos apenas através dos servidores de ficheiros - (*FSVROONLY especificado)
Só será efectuada uma pesquisa se aceder ao iSeries através de um servidor de ficheiros. Se esta opção não estiver seleccionada, serão pesquisados todos os acessos.
- Falha no pedido se falhar o programa de saída - (*ERRFAIL especificado)
Se existirem erros quando for chamado o programa de saída, o pedido ou operação que accionou a chamada do programa de saída falhará. Se esta opção não estiver seleccionada, o sistema ignorará a falha do programa de saída e o objecto será considerado como se não tivesse sido pesquisado.
- Executar actualizações do acesso para escrita - (*NOWRTUPG não especificado)
A actualização do acesso ocorrerá para o descritor da pesquisa transferido para o programa de saída, de forma a incluir o acesso para escrita. Se a opção *NOWRTUPG não for seleccionada, o sistema não efectuará tentativas de actualização do acesso para escrita.
Se a opção *NOWRTUPG for especificada, o sistema **não** tentará actualizar o acesso para o descritor da pesquisa transferido para o programa de saída, de forma a incluir o acesso para escrita. Se a opção *NOWRTUPG não for especificada, o sistema tentará efectuar a actualização do acesso para escrita.
- Utilize o atributo 'só quando os objectos foram alterados' para controlar a pesquisa - (*USEOCOATR especificado)
Será utilizado o atributo 'só alteração de objecto' (só pesquisará o objecto se este tiver sido modificado). Se esta opção não for seleccionada, este atributo não será utilizado e o objecto será pesquisado depois de modificado e assim que o software de pesquisa indicar uma actualização.
- Falha no pedido de encerramento se a pesquisa falhar durante o encerramento - (*NOFAILCLO não especificado)
Se ocorreu uma falha na pesquisa de um objecto durante o processamento de encerramento, o pedido de encerramento falhará. Se esta opção não for seleccionada, o pedido de encerramento não falhará. Quando não seleccionada, este valor substitui a especificação do valor 'falha no pedido se falhar o programa de saída'.
Se a opção *NOFAILCLO for especificada, **não** ocorrerá um falha no pedido de encerramento com indicação de falha na pesquisa, mesmo que tenha ocorrido uma falha na pesquisa do objecto efectuada como parte do processamento de encerramento.
- Pesquisar no próximo acesso após restauro do objecto - (*NOPOSTRST não especificado)
Os objectos serão pesquisados após o restauro. Se o atributo 'o objecto não será pesquisado' estiver especificado, o objecto será pesquisado uma vez após o restauro. Se o atributo 'só alteração de objecto' estiver especificado, o objecto será pesquisado após o restauro.
Se a opção *NOPOSTRST for especificada durante o restauro dos objectos, estes não serão pesquisados apenas porque foram restaurados. Se o atributo de objecto for 'o objecto não será pesquisado', o objecto nunca será pesquisado. Se o atributo de objecto for 'só alteração de objecto', o objecto só será pesquisado se for modificado após o restauro.

Referências relacionadas

"Pesquisa no sistema de ficheiros integrado" na página 101

Os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros de utilizador UDFS ASP só serão pesquisados utilizando os pontos de saída relacionados com a pesquisa, quando os sistemas de ficheiros tiverem sido totalmente convertidos para o formato de directório *TYPE2.

Informações relacionadas

QSCANFS

QSCANFSCTL

Ocorrências de pesquisa

A pesquisa pode ocorrer por várias razões. Seguem-se algumas informações sobre quando e por que motivo poderá ocorrer uma pesquisa.

Para visualizar o estado actual da pesquisa e o atributo de um objecto, pode utilizar o comando Trabalhar com ligações de objectos (WRKLNK), o comando Apresentar ligações de objectos (DSPLNK), a API Obter atributos (QP01GetAttr()) ou a página Propriedades (Properties) no iSeries Navigator.

Informações relacionadas

Comando Trabalhar com ligações de objectos (WRKLNK)

Comando Apresentar ligações de objectos (DSPLNK)

API Obter atributos (QP01GetAttr())

Alteração de objecto:

Poderá ser efectuada uma pesquisa se o objecto for acedido após ter sido alterado ou modificado.

Normalmente, a modificação ocorre nos dados do objecto. Modificações efectuadas a um objecto são, por exemplo, escrever no objecto directamente ou através da definição de memória, truncar ou eliminar o objecto. Se o atributo CCSID do objecto for alterado, será efectuada uma pesquisa no próximo acesso.

Alteração de assinatura:

Será efectuada uma pesquisa quando o objecto for acedido se a assinatura global for diferente da assinatura do objecto.

As assinaturas globais ou de grupo ASP independente representam o nível de software associado aos programas de saída relacionados com a pesquisa. A assinatura do objecto reflecte a assinatura global ou de ASP independente da última pesquisa efectuada ao objecto. Quando um objecto não se encontra num grupo de ASPs independentes, a assinatura do objecto é comparada com a assinatura de pesquisa global. Se o objecto se encontrar num ASP independente, a assinatura do objecto é comparada com a assinatura de pesquisa de grupo de ASPs independentes associado.

Nota: No exemplo que se segue, são utilizadas chaves de pesquisa de expressões e assinaturas chave de pesquisa. A chave de pesquisa é um método de identificação de um conjunto de software de pesquisa. Um exemplo disto é o de uma empresa específica. A assinatura chave de pesquisa permite que o conjunto de software de pesquisa indique o nível de suporte fornecido. Um exemplo disto é um conjunto de definições de vírus.

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

1. Um programa de saída é registado no ponto de saída QIBM_QP0L_SCAN_OPEN. Foram especificadas uma chave e uma assinatura chave de pesquisa da seguinte forma:

Chave de pesquisa: XXXXXX
Assinatura chave de pesquisa: 0000000000

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

2. Um programa de saída é então registado no ponto de saída QIBM_QP0L_SCAN_CLOSE. Foram especificadas uma chave e uma assinatura chave de pesquisa da seguinte forma:

Chave de pesquisa: XXXXXX
Assinatura chave de pesquisa: 1111111111

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

3. A seguir, é aberto um ficheiro que tem actualmente uma assinatura de objecto de 0000. A existência dos programas de saída, juntamente com a diferença de assinaturas de pesquisa globais (0000 a 0001), inicia uma pesquisa. Depois de concluída a pesquisa com êxito, a assinatura do ficheiro é actualizada para 0001.

4. Se o ficheiro for aberto por outro utilizador, não será efectuada nova pesquisa, uma vez que as assinaturas do objecto e as assinaturas globais correspondem.

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

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

Chave de pesquisa: XXXXXX
Assinatura chave de pesquisa: 2222222222

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

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

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

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

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

Informações relacionadas

QIBM_QP0L_SCAN_OPEN
QIBM_QP0L_SCAN_CLOSE
API Alterar assinatura de pesquisa (QP0LCHSG)

CCSID diferente:

Se um objecto for acedido com um identificador do conjunto de caracteres codificados (CCSID - Coded Character Set Identifier) diferente do identificador pesquisado anteriormente para esse objecto, é activada uma pesquisa.

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

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

Durante uma operação de salvaguarda:

Esta situação é ainda um outro exemplo de quando pode ocorrer uma pesquisa. Uma pesquisa pode ser pedida quando um objecto é guardado.

O comando Guardar objecto (SAV) inclui agora um parâmetro SCAN que permite especificar se os ficheiros serão pesquisados ao serem guardados. Também pode solicitar que o objecto não seja guardado se anteriormente falhou numa pesquisa ou se falhar durante uma pesquisa ao guardar. Deste modo, evita que os ficheiros que falharam na pesquisa sejam colocados nos suportes de dados e possivelmente movidos para outros sistemas.

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

Informações relacionadas

Comando Guardar objecto (SAV)

Verificar a integridade do objecto:

Por último, pode ser solicitada uma pesquisa se o parâmetro SCANFS do comando Verificar integridade do objecto (CHKOBJITG) for especificado juntamente com um valor *YES.

Esta opção será a ideal se pretende determinar se um ficheiro está em bom estado sem ter de o abrir. Se SCANFS (*STATUS) for especificado, todos os objectos com falha nas pesquisas anteriores terão um registo de violação de falha de pesquisa.

Informações relacionadas

Comando Alterar integridade do objecto (CHGOBJITG)

Trabalhar com sistemas de ficheiros

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

Cada sistema de ficheiros tem um conjunto de estruturas e regras lógicas para interagir com as informações em memória. Estas estruturas e regras podem ser diferentes de um sistema de ficheiros para outro. De facto, na perspectiva das estruturas e regras, o suporte do i5/OS para aceder a ficheiros de base de dados e a vários outros tipos de objectos através de bibliotecas pode ser considerado como um sistema de ficheiros. Do mesmo modo, o suporte do i5/OS para aceder a documentos (que são, na realidade, ficheiros de dados contínuos) através da estrutura de pastas pode ser considerado como um sistema de ficheiros separado.

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

Pode interagir com qualquer um dos sistemas de ficheiros através de uma interface comum. Esta interface é optimizada para a entrada e saída de dados de sequência, ao contrário da entrada e saída de registos, que é fornecida através das interfaces de gestão de dados. Os comandos, menus, ecrãs e as interfaces de programas de aplicações (APIs) fornecidos permitem a interacção com os sistemas de ficheiros através desta interface comum.

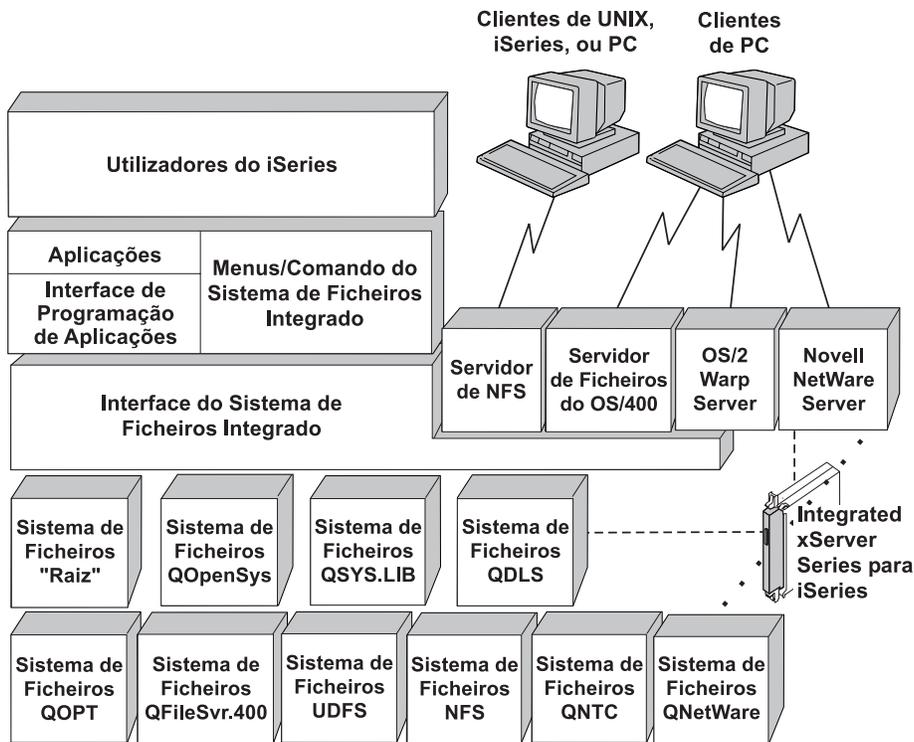


Figura 9. Sistemas de ficheiros, servidores de ficheiros e interface do sistema de ficheiros integrado

Utilizar o Network File System através da interface do sistema de ficheiros integrado

É possível aceder ao Network File System (NFS) através da interface do sistema de ficheiros integrado. Tenha em atenção estas considerações e limitações.

Conceitos relacionados

“Em que consiste o sistema de ficheiros integrado” na página 2

O sistema de ficheiros integrado é uma parte do i5/OS que suporta a entrada/saída de sequência e gestão do armazenamento semelhante ao computador pessoal e aos sistemas operativos UNIX, fornecendo ao mesmo tempo uma estrutura de integração de todas as informações armazenadas no servidor.

Tarefas relacionadas

“Aceder utilizando menus e ecrãs” na página 72

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

“Mover ficheiros ou pastas para outro sistema de ficheiros” na página 137

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 estejam actualmente armazenados. Poderá pretender mover objectos de um sistema de ficheiros para outro para tirar partido dessas características.

Referências relacionadas

“Aceder utilizando comandos de CL” na página 74

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através da interface do sistema de ficheiros integrado.

“Regras de nomes de caminhos para comandos de CL e ecrãs” na página 77

Ao utilizar um comando ou ecrã do sistema de ficheiros integrado para operar um objecto, este é identificado fornecendo o respectivo nome de caminho.

“Executar operações utilizando APIs” na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Informações relacionadas

Suporte óptico

Suporte de Network file system

Comparação entre sistemas de ficheiros

Estas tabelas resumem as funções e limitações de cada sistema de ficheiros.

Tabela 2. Resumo do sistema de ficheiros (Parte 1 de 2)

Capacidade	"raiz" (/)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Parte standard do i5/OS	Sim	Sim	Sim	Sim	Sim
Tipo de ficheiro	Dados Contínuos	Dados Contínuos	Registo ¹²	Dados Contínuos	Dados Contínuos
Limite do tamanho do ficheiro	T2=1 TB; T1=128 GB	T2=1 TB; T1=128 GB	Tamanhos dos ficheiros de base de dados	4 GB	Varia ¹⁷
Integrado com o OfficeVision [®] (por exemplo, o ficheiro pode ser enviado por correio)	Não	Não	Não	Sim	Não
Acesso através do servidor de ficheiros do i5/OS	Sim	Sim	Sim	Sim	Sim
Acesso directo através do processador de E/S do servidor de ficheiros ¹	Não	Não	Não	Não	Sim
Velocidade comparativa para abrir/fechar	Média ²	Média ²	Baixa ²	Baixa ²	Média ²
Procura de nomes sensível a maiúsculas e minúsculas	Não	Sim	Não ⁴	Não ⁵	Não
Comprimento máximo de cada componente no nome de caminho	255 car ¹⁹	255 car ¹⁹	10,6 car ⁶	8,3 car ⁷	255 car ¹⁹
Comprimento máximo do nome de caminho ⁸	16MB	16MB	55 – 66 car ⁴	82 car	255 car
Comprimento máximo dos atributos expandidos para um objecto	2GB	2GB	Varia ⁹	32KB	0 ¹⁸
Níveis máximos da hierarquia de directórios dentro do sistema de ficheiros	Sem limite ¹⁰	Sem limite ¹⁰	3	32	127
Máximo de ligações por objecto ¹¹	Varia ¹⁵	Varia ¹⁵	1	1	1
Suporta ligações simbólicas	Sim	Sim	Não	Não	Não
Objecto/ficheiro pode ter proprietário	Sim	Sim	Sim	Sim	Não
Suporta os comandos do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta as APIs do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim

Tabela 2. Resumo do sistema de ficheiros (Parte 1 de 2) (continuação)

Capacidade	"raiz" (/)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Suporta APIs do sistema de ficheiros hierárquico (HFS)	Não	Não	Não	Sim	Não
Protecção por módulo ¹³	Sim	Sim	Sim	Não	Sim
Suporta registo em diário de objectos	Sim	Sim	Sim ¹⁴	Não	Não

Notas:

1. O processador de E/S do servidor de ficheiros é hardware utilizado pelo servidor de Rede Local (LAN).
2. Quando acedido através do servidor de ficheiros do i5/OS.
3. Quando acedido através de um PC cliente do servidor de Rede Local (LAN). O acesso através das APIs do iSeries é comparativamente mais lento.
4. O sistema de ficheiros QSYS.LIB tem um comprimento de nome de caminho máximo de 55 caracteres. O sistema de ficheiros QSYS.LIB de ASP independente tem um comprimento de caminho máximo de 66 caracteres.
5. Consulte "Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)" na página 52 para obter detalhes.
6. Até 10 caracteres para o nome do objecto e até 6 caracteres para o tipo do objecto.
7. Até 8 caracteres para o nome e de 1 a 3 caracteres para a extensão do tipo de ficheiro (se existir alguma).
8. Assumindo um nome de caminho absoluto que comece por / seguido do nome do sistema de ficheiros (como, por exemplo, /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 nos sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente suportam entrada e saída 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 suportam o registo em diário de tipos de objectos diferentes dos sistemas de ficheiros "raiz" (/), UDFS e QOpenSys.
15. Os directórios *TYPE2 têm um limite de um milhão de ligações por objecto e um limite de 999 9989 subdirectórios. Os directórios *TYPE1 têm um limite de 32 767 ligações por objecto.
16. Os dados desta coluna referem-se a ambos os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente.
17. Depende do sistema a ser acedido.
18. O QNTC não suporta atributos expandidos.
19. Para determinados valores CCSID, o comprimento máximo pode ser inferior a 255 caracteres.

Abreviaturas

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

Tabela 3. Resumo do sistema de ficheiros (Parte 2 de 2)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Parte standard do i5/OS	Sim	Sim	Sim	Sim	Não

Tabela 3. Resumo do sistema de ficheiros (Parte 2 de 2) (continuação)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Tipo de ficheiro	Dados Contínuos				
Limite do tamanho do ficheiro	4 GB	2 GB - 1	T2 = 1 TB; T1=128 GB	Varia ¹⁶	2 GB
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 i5/OS	Sim	Sim	Sim	Sim	Sim
Acesso directo através do Integrated PC Server ¹	Não	Não	Não	Não	Sim
Velocidade comparativa para abrir/fechar	Baixa	Baixa ²	Média ²	Média ²	Alta ¹¹
Procura de nomes sensível a maiúsculas e minúsculas	Não	Não ²	Sim ¹²	Varia ²	Não
Comprimento máximo de cada componente no nome de caminho	Varia ⁴	Varia ²	255 car ¹⁷	Varia ²	255 car ^{13, 17}
Comprimento máximo do nome de caminho	294 car	Sem limite ²	16MB	Sem limite ²	255 car
Comprimento máximo dos atributos expandidos para um objecto	8MB	0 ⁶	2GB ¹⁰	0 ⁶	64KB
Níveis máximos da hierarquia de directórios dentro do sistema de ficheiros	Sem limite ⁷	Sem limite ²	Sem limite ⁷	Sem limite ²	100
Máximo de ligações por objecto ⁷	1	1	Varia ¹⁵	Varia ²	1
Suporta ligações simbólicas	Não	Não	Sim	Sim ²	Não
Objecto/ficheiro pode ter proprietário	Não	Não ⁹	Sim	Sim ²	Sim
Suporta os comandos do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta as APIs do sistema de ficheiros integrado	Sim	Sim	Sim	Sim	Sim
Suporta APIs do sistema de ficheiros hierárquico (HFS)	Sim	Não	Não	Não ²	Não
Protecção por módulo ¹⁴	Sim	Sim	Sim	Sim	Não
Suporta registo em diário de objectos	Não	Não	Sim	Não	Não

Tabela 3. Resumo do sistema de ficheiros (Parte 2 de 2) (continuação)

Capacidade	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Notas:					
<ol style="list-style-type: none"> 1. O processador de E/S do servidor de ficheiros é hardware utilizado pelo servidor de Rede Local (LAN). 2. Depende em que sistema de ficheiros está a ser acedido. 3. Quando acedido através do servidor de ficheiros do i5/OS. 4. Consulte "Sistema de ficheiros óptico (QOPT)" na página 54 para obter detalhes. 5. Assumindo um nome de caminho absoluto que comece por / seguido do nome do sistema de ficheiros. 6. O sistema de ficheiros QFileSvr.400 não devolve atributos expandidos mesmo que o sistema de ficheiros a ser acedido suporte atributos expandidos. 7. Na prática, os níveis de directórios são limitados pelos limites de espaço do programa e do sistema. 8. Excepto um directório, que pode ter apenas uma ligação a outro directório. 9. O sistema de ficheiros a ser acedido poderá suportar proprietários de objecto. 10. O comprimento máximo dos atributos expandidos para o próprio UDFS não pode exceder 40 bytes. 11. Quando acedido através de um PC cliente da Novell NetWare. O acesso através das APIs do iSeries é comparativamente mais lento. 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. 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. 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. 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. 16. Depende do sistema a ser acedido. 17. Para determinados valores CCSID, o comprimento máximo pode ser inferior a 255 caracteres. 					
Abreviaturas					
<ul style="list-style-type: none"> • car = caracteres • T1 = *TYPE1 *STMF • T2 = *TYPE2 *STMF • B = bytes KB = quilobytes MB = megabytes GB = gigabytes TB = terabytes 					

Tarefas relacionadas

"Mover ficheiros ou pastas para outro sistema de ficheiros" na página 137

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 estejam actualmente armazenados. Poderá pretender mover objectos de um sistema de ficheiros para outro para tirar partido dessas características.

Referências relacionadas

"sistema de ficheiros "raiz" (/)" na página 33

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. Apresenta as características dos sistemas de ficheiros do OS/2 e DOS.

"Sistema de ficheiros de sistemas abertos (QOpenSys)" na página 36

O sistema de ficheiros QOpenSys é compatível com as normas do sistema aberto com base em UNIX, como POSIX e X/Open Portability Guide (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.

“Sistemas de ficheiros definidos pelo utilizador (UDFSs)” na página 38

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System) residem no conjunto de memória auxiliar (ASP - Auxiliary Storage Pool) ou no conjunto de memória auxiliar (ASP) independente à sua escolha. Pode criar e gerir estes sistemas de ficheiros.

“Sistema de ficheiros de biblioteca (QSYS.LIB)” na página 45

O sistema de ficheiros QSYS.LIB suporta a estrutura de bibliotecas do servidor iSeries.

“QSYS.LIB de ASP independente” na página 48

O sistema de ficheiros QSYS.LIB de ASP independente suporta a estrutura de biblioteca do servidor iSeries em áreas de memória auxiliar (ASP - Auxiliary Storage Pool) independentes criadas e definidas pelo utilizador. Este sistema de ficheiros fornece acesso a ficheiros de base de dados e a todos os outros tipos de objectos do servidor iSeries que o suporte de biblioteca gere nos ASPs independentes.

“Sistema de ficheiros de serviços da biblioteca de documentos (QDLS)” na página 52

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

“Sistema de ficheiros óptico (QOPT)” na página 54

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

“Sistema de ficheiros NetWare (QNetWare)” na página 57

O sistema de ficheiros QNetWare fornece acesso a dados em servidores de PC autónomos a executar Novell NetWare 5.1 ou 6.0.

“Sistema de ficheiros do iSeries NetClient (QNTC)” na página 60

O sistema de ficheiros QNTC fornece acesso a dados e objectos armazenados num Servidor xSeries Integrado para iSeries com o Windows NT 4.0 Server ou posterior ou Linux[®]. O sistema de ficheiros QNTC também fornece acesso a dados e objectos armazenados em servidores remotos com o Windows NT 4.0 ou posterior, Linux Samba 3.0 ou posterior, ou versões suportadas do iSeries NetServer[™].

“Sistema de ficheiros do servidor de ficheiros do i5/OS (QFileSvr.400)” na página 65

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

“Network File System (NFS)” na página 69

O NFS (Network File System) permite ao utilizador aceder a dados e objectos armazenados num servidor de NFS remoto.

Informações relacionadas

Gestão de registos em diário

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. Apresenta as características dos sistemas de ficheiros do OS/2 e DOS.

Para além disso:

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

O sistema de ficheiros "raiz" (/) tem suporte para os ficheiros especiais de caracteres (*CHRFS) denominados /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 *CHRFS 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.

Utilizar o sistema de ficheiros "raiz" (/)

É possível aceder ao sistema de ficheiros "raiz" (/) através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros do i5/OS ou as APIs, ecrãs de utilizador e comandos do sistema de ficheiros integrado.

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 caminhos no sistema de ficheiros "raiz" (/)

Os nomes de caminhos têm um formato específico no sistema de ficheiros "raiz" (/).

| /Directório/Directório . . . /Objecto

- Cada componente do nome de caminho pode ter até 255 caracteres de comprimento, muito mais do que nos sistemas de ficheiros QSYS.LIB ou QDLS. O nome completo do caminho pode ser extremamente longo, até 16 megabytes.
- Não existe limite para a profundidade da hierarquia de directórios, para além dos limites de espaço do programa e servidor.
- Os caracteres dos nomes são convertidos no formato UCS2 Nível 1 (para directórios *TYPE1) e UTF-16 (para directórios *TYPE2) quando os nomes são armazenados.

Conceitos relacionados

"Continuidade de nomes" na página 19

Ao utilizar os sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

"Directórios *TYPE2" na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

"Nome de caminho" na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

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 ligar um sistema de ficheiros "raiz" (/) a um objecto de outro sistema de ficheiros, como, por exemplo, QSYS.LIB, QSYS.LIB de ASP independente ou QDLS.

Conceitos relacionados

"Ligação" na página 12

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.

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

Todos os comandos listados no tópico "Aceder utilizando comandos de CL" e os ecrãs descritos no tópico "Aceder utilizando menus e ecrãs" podem operar no sistema de ficheiros "raiz" (/). No entanto, a utilização destes comandos num processo que suporte vários módulos pode não oferecer segurança.

Tarefas relacionadas

"Aceder utilizando menus e ecrãs" na página 72

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

Referências relacionadas

"Aceder utilizando comandos de CL" na página 74

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através da interface do sistema de ficheiros integrado.

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

Todas as APIs listadas no tópico "Executar operações utilizando APIs" podem operar no sistema de ficheiros "raiz" (/).

Referências relacionadas

"Executar operações utilizando APIs" na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Registar em diário alterações a objectos no sistema de ficheiros "raiz" (/)

Os objectos no sistema de ficheiros "raiz" (/) podem ser registados em diário. Esta função permite recuperar as alterações efectuadas num objecto que tenham ocorrido desde a última vez que o objecto tenha sido guardado.

Conceitos relacionados

"Registar objectos em diário" na página 102

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

Dispositivos UDP e TCP no sistema de ficheiros "raiz" (/)

O sistema de ficheiros "raiz" (/) no directório /dev/xti suportará agora dois controladores de dispositivos denominados udp e tcp.

Estes dois controladores são ficheiros especiais de caracteres (*CHRSEs) e são criados durante o primeiro carregamento do programa inicial (IPL - Initial Program Load). Os controladores de dispositivos UDP e TCP são utilizados para abrir uma ligação para os fornecedores de transporte UDP e TCP. Estes dois controladores são dispositivos de utilizador e recebem um número principal de novo dispositivo. Também têm aberturas idênticas, o que significa que cada abertura obtém uma instância única do dispositivo. A utilização destes dispositivos apenas é suportada no PASE (Portable Application Solutions Environment) do i5/OS. A tabela que se segue contém os objectos que serão criados, bem como as respectivas propriedades.

Tabela 4. Objectos e propriedades do controlador de dispositivos

Nome de caminho	Tipo	Principal	Secundário	Propriet.	Autorid. sobre os dados do propriet.	Grupo	Autorid. sobre os dados do grupo	Autorid. sobre os dados públicos
/dev/xti	*DIR	N/A	N/A	QSYS	*RWX	Nenhum	*RX	*RX
/dev/xti/tcp	*CHRFS	Conjunto de unidades	TCP	QSYS	*RW	Nenhum	*RW	*RW
/dev/xti/udp	*CHRFS	Conjunto de unidades	UDP	QSYS	*RW	Nenhum	*RW	*RW

Conceitos relacionados

“Directórios fornecidos” na página 7

Ao reiniciar o sistema, o sistema de ficheiros integrado cria estes directórios, caso ainda não existam.

Informações relacionadas

PASE (Portable Applications Solutions Environment)

Sistema de ficheiros de sistemas abertos (QOpenSys)

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

Para além disso:

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

O sistema de ficheiros QOpenSys tem as mesmas características do sistema de ficheiros “raiz” (/), à excepção da sensibilidade a maiúsculas e minúsculas que permite activar o suporte das normas de sistemas abertos baseados em UNIX.

Utilizar QOpenSys

É possível aceder a QOpenSys através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros do i5/OS ou as APIs, ecrãs de utilizador e comandos do sistema de ficheiros integrado.

Conceitos relacionados

“Directórios fornecidos” na página 7

Ao reiniciar o sistema, o sistema de ficheiros integrado cria estes directórios, caso ainda não existam.

Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QOpenSys

Ao contrário do sistema de ficheiros “raiz” (/), o sistema de ficheiros QOpenSys efectua 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 um objecto denominado Payro11, um objecto denominado PayRo11 e um objecto denominado PAYROLL no mesmo directório de QOpenSys.

Nomes de caminhos no sistema de ficheiros QOpenSys

Os nomes de caminhos têm um formato específico no sistema de ficheiros QOpenSys.

| /QOpenSys/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 MB.
- 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.

Conceitos relacionados

“Continuidade de nomes” na página 19

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

“Directórios *TYPE2” na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Ligações no sistema de ficheiros QOpenSys

São permitidos vários caminhos directos 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.

Conceitos relacionados

“Ligação” na página 12

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.

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

Todos os comandos listados no tópico “Aceder utilizando comandos de CL” e os ecrãs descritos no tópico “Aceder utilizando menus e ecrãs” podem operar no sistema de ficheiros QOpenSys. No entanto, a utilização destes comandos num processo que suporte múltiplos módulos pode não ser segura.

Tarefas relacionadas

“Aceder utilizando menus e ecrãs” na página 72

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

Referências relacionadas

“Aceder utilizando comandos de CL” na página 74

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser

efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através da interface do sistema de ficheiros integrado.

Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QOpenSys

Todas as APIs listadas no tópico "Executar operações utilizando APIs" podem operar no sistema de ficheiros QOpenSys.

Referências relacionadas

"Executar operações utilizando APIs" na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Registar em diário alterações a objectos no sistema de ficheiros QOpenSys

Os objectos no sistema de ficheiros QOpenSys podem ser registados em diário. Esta função permite recuperar as alterações efectuadas num objecto que tenham ocorrido desde a última vez que o objecto tenha sido guardado.

Conceitos relacionados

"Registar objectos em diário" na página 102

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

Sistemas de ficheiros definidos pelo utilizador (UDFSs)

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System) residem no conjunto de memória auxiliar (ASP - Auxiliary Storage Pool) ou no conjunto de memória auxiliar (ASP) independente à sua escolha. Pode criar e gerir estes sistemas de ficheiros.

Além disso, estes sistemas:

- Fornecem uma estrutura de directórios hierárquica, semelhante à dos sistemas operativos de PC, tal como DOS e OS/2
- Estão optimizados para entrada e saída de ficheiros de dados contínuos
- Suportam vários caminhos directos e ligações simbólicas
- Suportam terminais locais
- Suportam APIs protegidas por módulos
- Suportam objectos *FIFO
- Suportam o registo de alterações de objectos
- Suportam a pesquisa de objectos através dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado

Pode criar múltiplos UDFSs atribuindo a cada um nome exclusivo. Pode especificar outros atributos para um UDFS durante a criação, incluindo:

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

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

- O atributo de pesquisa de objectos criados que define qual o atributo de pesquisa para objectos criados num UDFS.
- O atributo de restrição de mudança de nome e de remoção.
- O valor de auditoria para um UDFS.
- Os diferentes formatos de ficheiros de dados contínuos, *TYPE1 e *TYPE2.

Conceitos sobre o sistema de ficheiros definido pelo utilizador

Num sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), tal como nos sistemas de ficheiros “raiz” (/) e QOpenSys, pode criar directórios, ficheiros de dados contínuos, ligações simbólicas, terminais locais e objectos *FIFO.

Um objecto ficheiro especial de um só bloco (*BLKSF) representa um UDFS. À medida que cria UDFSs, também cria automaticamente ficheiros especiais de bloco. O utilizador só pode ter acesso ao ficheiro especial de blocos utilizando os comandos genéricos do sistema de ficheiros integrado, as APIs e a interface 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). Ao instalar um UDFS num directório, o conteúdo original desse directório, incluindo objectos e subdirectórios, fica inacessível. Quando instala um UDFS, o seu conteúdo fica acessível através do caminho do directório sobre o qual instalou o UDFS. Por exemplo, o directório /home/JOSE contém um ficheiro /home/JOSE/salarios. Um UDFS contém três directórios correio, accao e enviar. Depois de instalar o UDFS em /home/JOSE, o ficheiro /home/JOSE/salarios fica inacessível e os três directórios do UDFS ficam acessíveis como /home/JOSE/correio, /home/JOSE/accao e /home/JOSE/enviar. Depois de desinstalar o UDFS, o ficheiro /home/JOSE/salarios fica de novo acessível e os três directórios do UDFS ficam inacessíveis. Um carregamento do programa inicial (IPL - Initial Program Load) do sistema desmonta todos os UDFSs. Deste modo, é necessário montar novamente os UDFSs após qualquer IPL.

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

Para obter mais informações sobre como montar sistemas de ficheiros, consulte Suporte do Network File System no OS/400 .

Utilizar o sistema de ficheiros definido pelo utilizador através da interface do sistema de ficheiros integrado

É possível aceder ao sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros do i5/OS ou as APIs, ecrãs de utilizador e comandos 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.

Conceitos relacionados

“Ligação” na página 12

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.

“Ficheiro de dados contínuos” na página 18

Um *ficheiro de dados contínuos* é uma sequência de bytes acessível aleatoriamente, sem qualquer estrutura adicional imposta pelo sistema.

Informações relacionadas

Comando Criar UDFS (CRTUDFS)

Sensibilidade a maiúsculas e minúsculas num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

Pode especificar se os nomes de objecto no sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) serão ou não sensíveis a maiúsculas e minúsculas ao criar o mesmo.

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

Quando activa (a opção) insensibilidade a maiúsculas e minúsculas, o servidor não distinguirá entre maiúsculas e minúsculas durante a procura de nomes. Deste modo, o servidor reconhecerá /home/CARLA e /HOME/carla como o mesmo directório e não como dois directórios diferentes. Para criar um UDFS sensível 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.

Informações relacionadas

Comando Criar sistema de ficheiros definido pelo utilizador (CRTUDFS)

Nomes de caminhos num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

Um ficheiro especial de blocos (*BLKSF) representa um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) 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 MB.

- 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.

Conceitos relacionados

“Continuidade de nomes” na página 19

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

“Directórios *TYPE2” na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), “raiz” (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Ligações num sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

Um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) permite vários caminhos directos para o mesmo objecto e suporta totalmente as ligações simbólicas.

Uma ligação simbólica pode criar uma ligação de um UDFS a um objecto noutra sistema de ficheiros.

Conceitos relacionados

“Ligação” na página 12

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.

Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros definido pelo utilizador

Todos os comandos listados no tópico “Aceder utilizando comandos de CL” e os ecrãs descritos no tópico “Aceder utilizando menus e ecrãs” podem operar num sistema de ficheiros definido pelo utilizador.

Existem alguns comandos de CL específicos para o UDFS e outros sistemas de ficheiros montados em geral. A tabela seguinte descreve-os.

Tabela 5. Comandos de CL do sistema de ficheiros definido pelo utilizador

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

Tabela 5. Comandos de CL do sistema de ficheiros definido pelo utilizador (continuação)

Comando	Descrição
UNMOUNT	Desinstalar um Sistema de Ficheiros. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local. Este comando é um nome alternativo correspondente ao comando RMVMFS.

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

Tarefas relacionadas

“Aceder utilizando menus e ecrãs” na página 72

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

Referências relacionadas

“Aceder utilizando comandos de CL” na página 74

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através da interface do sistema de ficheiros integrado.

Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros definido pelo utilizador

Todas as APIs listadas no tópico “Executar operações utilizando APIs” podem operar num sistema de ficheiros definido pelo utilizador.

Nota: Tem de montar um UDFS para poder executar APIs do sistema de ficheiros integrado nos objectos armazenados nesse UDFS.

Referências relacionadas

“Executar operações utilizando APIs” na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Interface gráfica do utilizador para um sistema de ficheiros definido pelo utilizador

O iSeries Navigator, uma interface gráfica de utilizador no PC, fornece um acesso fácil e conveniente a sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System).

Esta interface permite-lhe criar, eliminar, visualizar, instalar e desinstalar um UDFS de um cliente do Windows.

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

- “Criar um novo sistema de ficheiros definido pelo utilizador” na página 141
- “Montar sistema de ficheiros definido pelo utilizador” na página 141
- “Desmontar sistema de ficheiros definido pelo utilizador” na página 141

Criar um sistema de ficheiros definido pelo utilizador no 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 nomes, das APIs e dos comandos de CL do sistema de ficheiros integrado.

Os comandos ADDMFS ou MOUNT colocam o sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) no início do directório local já existente. Pode criar um UDFS num ASP ou ASP independente que escolher.

Também pode especificar os seguintes itens para um UDFS:

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

Informações relacionadas

Comando Criar sistema de ficheiros definido pelo utilizador (CRTUDFS)

Comando Adicionar sistema de ficheiros montado (ADDMFS)

Eliminar um sistema de ficheiros definido pelo utilizador do sistema de ficheiros integrado

O comando Eliminar sistema de ficheiros definido pelo utilizador (DLTUDFS) permite eliminar um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), não montado existente, bem como todos os objectos contidos no mesmo.

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.

Informações relacionadas

Comando Eliminar sistema de ficheiros definido pelo utilizador (DLTUDFS)

Apresentar um sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

O comando Apresentar sistema de ficheiros definido pelo utilizador (DSPUDFS) [Display User-Defined File System (DSPUDFS)] apresenta os atributos de um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) existente, montado ou não montado.

O comando Apresentar informações sobre sistemas de ficheiros montados (DSPMFSINF) também apresenta informações sobre um UDFS montado, bem como sobre qualquer sistema de ficheiros montado.

Informações relacionadas

Comando Apresentar sistema de ficheiros definido pelo utilizador (DSPUDFS)

Comando Apresentar informações do sistema de ficheiros montado (DSPMFSINF)

Montar um sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

Os comandos Adicionar sistema de ficheiros montado (ADDMFS) e MOUNT tornam os objectos de um sistema de ficheiros acessíveis ao espaço de nomes do sistema de ficheiros integrado.

Para montar um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), tem de especificar *UDFS para o parâmetro TYPE no comando ADDMFS.

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

Informações relacionadas

Comando Adicionar sistema de ficheiros montado (ADDMFS)

Desmontar um sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

O comando para desmontar torna o conteúdo de um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) inacessível às 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 tornam um sistema de ficheiros montado inacessível ao espaço de nomes do sistema de ficheiros integrado. Se qualquer dos objectos do sistema de ficheiros estiver a ser utilizado (por exemplo, se um ficheiro estiver aberto) quando o comando estiver a ser executado, será apresentada uma mensagem de erro. O UDFS permanecerá instalado. Se tiver feito a instalação sobre alguma parte do UDFS, este último UDFS não poderá ser desinstalado até ser "descoberto".

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

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

Informações relacionadas

Comando Remover sistema de ficheiros montado (RMVMFS)

Guardar e restaurar um sistema de ficheiros definido pelo utilizador no sistema de ficheiros integrado

O utilizador pode guardar e restaurar todos os objectos do sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), bem como as respectivas autoridades associadas.

O comando Guardar Objecto (SAV) permite guardar objectos num UDFS, ao passo que o comando Restaurar Objecto (RST) permite restaurar objectos do UDFS. Ambos os comandos podem ser executados, quer o UDFS esteja montado ou não. No entanto, para guardar correctamente os atributos do UDFS e não apenas os objectos do UDFS, o UDFS deverá ser desinstalado.

Informações relacionadas

Comando Guardar objecto (SAV)

Comando Restaurar objecto (RST)

Registar em diário alterações a objectos no sistema de ficheiros definido pelo utilizador

Os objectos nos sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems) podem ser registados em diário. Esta função permite recuperar as alterações efectuadas num objecto que tenham ocorrido desde a última vez que o objecto tenha sido guardado.

Conceitos relacionados

"Registar objectos em diário" na página 102

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

Sistema de ficheiros definido pelo utilizador e áreas de memória auxiliar (ASP) independentes

Ao efectuar a activação numa ASP independente, ocorrem várias alterações no sistema de ficheiros "raiz" (/).

As alterações em questão são as seguintes:

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

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

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

Referências relacionadas

“QSYS.LIB de ASP independente” na página 48

O sistema de ficheiros QSYS.LIB de ASP independente suporta a estrutura de biblioteca do servidor iSeries em áreas de memória auxiliar (ASP - Auxiliary Storage Pool) independentes criadas e definidas pelo utilizador. Este sistema de ficheiros fornece acesso a ficheiros de base de dados e a todos os outros tipos de objectos do servidor iSeries que o suporte de biblioteca gere nos ASPs independentes.

Sistema de ficheiros de biblioteca (QSYS.LIB)

O sistema de ficheiros QSYS.LIB suporta a estrutura de bibliotecas do servidor iSeries.

Este sistema de ficheiros fornece-lhe acesso a ficheiros de base de dados e a todos os outros tipos de objectos do servidor 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 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 a gestão de objectos do servidor iSeries
- Suporta operações de I/O de sequência em membros de ficheiros físicos, espaços de utilizador e ficheiros de salvaguarda.

Até à Versão 3 do i5/OS, é provável 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 serviços, como DDS, para desenvolver aplicações, utilizavam o sistema de ficheiros QSYS.LIB. Os operadores do sistema que utilizavam comandos, menus e ecrãs para manipular filas de saída utilizavam o sistema de ficheiros QSYS.LIB, bem como os administradores do sistema que criavam e alteravam os perfis dos utilizadores.

Todos estes serviços e as aplicações baseadas nos mesmos funcionam tal como funcionavam antes da introdução do sistema de ficheiros integrado. No entanto, estes serviços não conseguem aceder a 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 i5/OS ou as APIs, ecrãs de utilizador e comandos do sistema de ficheiros integrado.

Ao utilizar as interfaces do sistema de ficheiros integrado, deverá ter em conta as considerações e limitações que se seguem.

Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB

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

Restrições de processamento de ficheiros no sistema de ficheiros QSYS.LIB

Seguem-se algumas restrições a ter em consideração ao processar 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 contém um único campo, bem como ficheiros físicos origem, que contém 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 no intervalo de bytes, consulte o tópico fcntl().
- 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

QSYS.LIB suporta operações de entrada e saída de sequência para objectos de espaço 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 predefinido para o trabalho.

Suporte para ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB

O sistema de ficheiros QSYS.LIB suporta operações de I/O de sequência para objectos de ficheiros de salvaguarda.

Por exemplo, um ficheiro de salvaguarda existente tem dados que podem ser lidos ou copiados para outro ficheiro, até ser necessário colocá-los num objecto diferente de ficheiro de salvaguarda existente e vazio. Quando um ficheiro de salvaguarda é aberto para escrita, não são permitidas outras ocorrências abertas do ficheiro. Um ficheiro de salvaguarda **permite** várias ocorrências abertas para leitura, desde que nenhum trabalho tenha mais do que uma ocorrência aberta do ficheiro para leitura. Não é possível abrir

um ficheiro de salvaguarda para acesso de leitura/escrita. As operações de I/O de sequência em dados do ficheiro de salvaguarda não são permitidas quando estão a ser executados vários módulos num trabalho.

As operações de I/O de sequência num ficheiro de salvaguarda não são suportadas quando o referido ficheiro ou o respectivo directório estão a ser exportados através do servidor Network File System. 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 efectua a distinção entre maiúsculas e minúsculas nos 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 caminhos no sistema de ficheiros QSYS.LIB

Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto.

- Por exemplo:

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

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

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

- O nome do objecto em cada componente pode ter até 10 caracteres de comprimento e o tipo de objecto pode ter até 6 caracteres de comprimento.
- A hierarquia de directórios em QSYS.LIB pode ter dois ou três níveis (dois ou três componentes no nome de caminho), dependendo do tipo de objecto que estiver 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 primeiros dois níveis, a hierarquia de directórios de QSYS.LIB pode ter até cinco níveis.

- 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 CCSID, consulte o tópico Globalização do i5/OS.

Conceitos relacionados

"Nome de caminho" na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Ligações no sistema de ficheiros QSYS.LIB

Não é possível criar nem 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 processa a relação biblioteca-objecto como uma ligação. Deste modo, é possível efectuar a ligação entre um sistema de ficheiros que suporte ligações simbólicas e um objecto no sistema de ficheiros QSYS.LIB.

Conceitos relacionados

“Ligação” na página 12

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.

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

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QSYS.LIB.

Os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 podem operar no sistema de ficheiros QSYS.LIB, com as seguintes restrições:

- O comando ADDLNK só pode ser utilizado para criar uma ligação simbólica a um objecto em 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.
- O comando RCLLNK não é suportado.

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder utilizando menus e ecrãs” na página 72.

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

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QSYS.LIB.

As APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros QSYS.LIB, com as seguintes restrições:

- 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 efectuar a ligação a um objecto em QSYS.LIB a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- As APIs QjoStartJournal() e QjoEndJournal() não podem ser utilizadas em ficheiros físicos de base de dados.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

QSYS.LIB de ASP independente

O sistema de ficheiros QSYS.LIB de ASP independente suporta a estrutura de biblioteca do servidor iSeries em áreas de memória auxiliar (ASP - Auxiliary Storage Pool) independentes criadas e definidas pelo utilizador. Este sistema de ficheiros fornece acesso a ficheiros de base de dados e a todos os outros tipos de objectos do servidor 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 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 iSeries
- Suporta operações de I/O de sequência em membros de ficheiros físicos, espaços de utilizador e ficheiros de salvaguarda.

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 i5/OS ou as APIs, ecrãs de utilizador e comandos do sistema de ficheiros integrado.

Ao utilizar as interfaces do sistema de ficheiros integrado, deverá ter em conta as considerações e limitações que se seguem.

Referências relacionadas

“Sistema de ficheiros definido pelo utilizador e áreas de memória auxiliar (ASP) independentes” na página 44

Ao efectuar a activação numa ASP independente, ocorrem várias alterações no sistema de ficheiros “raiz” (/).

Lista de autorizações QPWFSEVER no sistema de ficheiros QSYS.LIB de ASP independente

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 de ASP independente que estejam 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 de ASP independente.

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

Restrições de processamento de ficheiros no sistema de ficheiros QSYS.LIB de ASP independente

Seguem-se as restrições a ter em consideração ao processar 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 no intervalo de bytes, consulte o tópico fcntl().
- 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

QSYS.LIB de ASP independente suporta operações de entrada e saída de sequência para objectos de espaço de utilizador.

Por exemplo, um programa pode escrever dados de sequência num espaço de utilizador e ler dados a partir de um espaço de utilizador. O tamanho máximo de um espaço de utilizador é de 16 776 704 bytes.

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

Suporte para ficheiros de salvaguarda no sistema de ficheiros QSYS.LIB de ASP independente

QSYS.LIB de ASP independente suporta operações de I/O de sequência para objectos de ficheiros de salvaguarda.

Por exemplo, um ficheiro de salvaguarda existente tem dados que podem ser lidos ou copiados para outro ficheiro, até ser necessário colocá-los num objecto diferente de ficheiro de salvaguarda existente e vazio. Quando um ficheiro de salvaguarda é aberto para escrita, não são permitidas outras ocorrências abertas do ficheiro. Um ficheiro de salvaguarda **permite** várias ocorrências abertas para leitura, desde que nenhum trabalho tenha mais do que uma ocorrência aberta do ficheiro para leitura. Não é possível abrir um ficheiro de salvaguarda para acesso de leitura/escrita. As operações de I/O de sequência em dados do ficheiro de salvaguarda não são permitidas quando estão a ser executados vários módulos num trabalho.

As operações de I/O de sequência num ficheiro de salvaguarda não são suportadas quando o referido ficheiro ou o respectivo directório estão a ser exportados através do servidor Network File System. 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 efectua a distinção entre maiúsculas e minúsculas nos 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 caminhos no sistema de ficheiros QSYS.LIB de ASP independente

Cada componente do nome de caminho tem de conter o nome do objecto seguido do tipo do objecto.

- Por exemplo:

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

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

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

- O nome do objecto em cada componente pode ter até 10 caracteres de comprimento e o tipo de objecto pode ter até 6 caracteres de comprimento.
- A hierarquia de directórios em QSYS.LIB de ASP independente pode ter dois ou três níveis (dois ou três componentes no nome de caminho), dependendo do tipo de objecto que estiver 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 /, nome_asp e QSYS.LIB forem incluídos como os três primeiros níveis, a hierarquia de directórios do sistema de ficheiros QSYS.LIB de ASP independente pode ter até seis níveis.

- 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 CCSID, consulte o tópico Globalização do i5/OS no iSeries Information Center.

Conceitos relacionados

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

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 processa a relação biblioteca-objecto como uma ligação. Deste modo, é possível efectuar a ligação entre um sistema de ficheiros que suporte ligações simbólicas e um objecto no sistema de ficheiros QSYS.LIB de ASP independente.

Conceitos relacionados

“Ligação” na página 12

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.

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

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QSYS.LIB de ASP independente.

Quase todos os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 podem operar no sistema de ficheiros QSYS.LIB de ASP independente. No entanto, existem algumas excepções:

- O comando ADDLNK só pode ser utilizado para criar uma ligação simbólica a um objecto em QSYS.LIB de ASP independente.
- As operações em ficheiros só podem ser efectuadas nos ficheiros físicos descritos pelo programa e nos ficheiros físicos origem.
- Os comandos STRJRN e ENDJRN não podem ser utilizados em ficheiros físicos de base de dados.
- Não pode mover bibliotecas do sistema de ficheiros QSYS.LIB de ASP independente para áreas de memória auxiliar (ASP - Auxiliary Storage Pool) básicas através do comando MOV. No entanto, pode mover bibliotecas de QSYS.LIB de ASP independente para a ASP do sistema ou para outras ASPs independentes.
- Se utilizar SAV ou RST para guardar ou restaurar objectos de biblioteca numa ASP independente, essa ASP independente tem de estar associada ao trabalho que executa a operação SAV ou RST ou a ASP independente tem de ser especificada no parâmetro ASPDEV. A convenção de nomenclatura de nomes de caminho/nome_asp/QSYS.LIB/tipo.objecto não é suportada em SAV e RST.
- O comando RCLLNK não é suportado.

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder utilizando menus e ecrãs” na página 72.

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

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QSYS.LIB de ASP independente.

As APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros QSYS.LIB de ASP independente, excepto nas seguintes situações:

- 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 efectuar a ligação a um objecto em 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.
- Se utilizar as APIs `QsrSave()` ou `QsrRestore()` para guardar ou restaurar objectos de biblioteca numa ASP independente, esta ASP independente tem de estar associada ao trabalho que executa a operação de salvaguarda ou restauro ou a ASP independente tem de ser especificada na chave ASPDEV. A convenção de nomenclatura do nome de caminho (`/nome_asp/QSYS.LIB/objecto.tipo`) não é suportada nas APIs `QsrSave()` e `QsrRestore()`.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

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

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

Para além disso:

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

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 i5/OS ou os comandos do sistema de ficheiros integrado, os ecrãs de utilizador e as APIs.

Ao utilizar as interfaces do sistema de ficheiros integrado, deverá ter em conta 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 de Objectos da Biblioteca de Documentos (DLO - Document Library Objects), mas também através da interface do sistema de ficheiros integrado ou das APIs fornecidas por um sistema hierárquico de ficheiros (HFS - Hierarchical File System).

Enquanto o sistema de ficheiros integrado se baseia no modelo do programa do Integrated Language Environment® (ILE), o HFS baseia-se no modelo de programa do servidor iSeries original.

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

Informações relacionadas

APIs do sistema hierárquico de ficheiros

Inscrição de utilizadores no sistema de ficheiros QDLS

Tem de estar inscrito no directório de distribuição do sistema ao trabalhar com objectos em QDLS.

Sensibilidade a maiúsculas e minúsculas no sistema de ficheiros QDLS

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.

Todos os outros caracteres são sensíveis a maiúsculas e minúsculas em QDLS.

Informações relacionadas

Nome do documento e da pasta

Nomes de caminhos no sistema de ficheiros QDLS

Cada componente do nome do caminho pode ser constituído apenas por um nome.

- Por exemplo:

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

Para obter mais informações sobre páginas de códigos, consulte o tópico Globalização do i5/OS no iSeries Information Center.

Conceitos relacionados

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Ligações no sistema de ficheiros QDLS

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

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

Conceitos relacionados

“Ligação” na página 12

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.

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

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QDLS.

Os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 podem operar no sistema de ficheiros QDLS, à excepção dos seguintes comandos:

- O comando ADDLNK só pode ser utilizado para efectuar a ligação a um objecto em 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 que se seguem não são suportados:
 - APYJRNCHG
 - CHGJRNOBJ
 - DSPJRN
 - ENDJRN
 - RCLLNK
 - RCVJRNE
 - RTVJRNE
 - SNDJRNE
 - STRJRN

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder utilizando menus e ecrãs” na página 72.

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

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QDLS.

As APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros QDLS, à excepção das seguintes APIs:

- A função symlink() só pode ser utilizada para efectuar a ligação a um objecto em QDLS a partir de outro sistema de ficheiros que suporte ligações simbólicas.
- Não são suportadas as seguintes funções:
 - givedescriptor()
 - ioctl()
 - link()
 - QjoEndJournal()
 - QjoRetrieveJournalEntries()
 - QjoRetrieveJournalInformation()
 - QJORJIDI()
 - QJOSJRNE()
 - QjoStartJournal()
 - Qp0lGetPathFromFileID()
 - readlink()
 - takedescriptor()

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Sistema de ficheiros óptico (QOPT)

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

Para além disso:

- Fornece uma estrutura de directórios hierárquica, semelhante à dos sistemas operativos de PC, tal como DOS e OS/2.

- Está optimizado para entrada e saída de ficheiros de dados contínuos.
- Suporta dados armazenados em ficheiros de dados contínuos.

Utilizar o QOPT através do sistema de ficheiros integrado

O sistema de ficheiros QOPT pode ser acedido através do sistema de ficheiros integrado utilizando o servidor de PC ou comandos, ecrãs de utilizador e APIs do sistema de ficheiros integrado.

Ao utilizar a interface do sistema de ficheiros integrado, deverá ter em conta as considerações e limitações que se seguem.

Informações relacionadas

Programação do dispositivo óptico

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 hierárquico de ficheiros (HFS - Hierarchical File System).

Enquanto o sistema de ficheiros integrado se baseia no modelo do programa do Integrated Language Environment (ILE), o HFS baseia-se no modelo de programa do servidor iSeries original.

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

Para obter mais informações sobre APIs de HFS, consulte a publicação Programação do dispositivo óptico. 

Informações relacionadas

APIs do sistema hierárquico de ficheiros

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 em QOPT. Contudo, as procuras de ficheiros e directórios são sensíveis a maiúsculas e minúsculas, independentemente do formato do suporte óptico.

Nomes de caminhos no sistema de ficheiros QOPT

O nome de caminho tem de começar por uma barra (/). O caminho é constituído pelo nome do sistema de ficheiros, pelo nome do volume, pelos nomes de directórios e subdirectórios e pelo nome do ficheiro.

- Por exemplo:

/QOPT/NOMEVOLUME/NOMEDIRECTÓRIO/NOMESUBDIRECTÓRIO/NOMEFICH

- É necessário o nome do sistema de ficheiros, QOPT.
- O volume e o comprimento do nome do caminho variam conforme o formato do suporte óptico.
- Pode especificar /QOPT no nome do caminho ou incluir um ou mais directórios ou subdirectórios no nome de caminho. Os nomes de directórios e de ficheiros permitem a utilização de qualquer carácter, excepto de X'00' a X'3F', X'FF'. Podem ser aplicáveis restrições adicionais com base no formato do suporte óptico.
- O nome do ficheiro é o último elemento no nome de caminho. O comprimento do nome do ficheiro é limitado pelo comprimento do nome do directório no caminho.

Para obter mais detalhes sobre regras de nomes de caminho no sistema de ficheiros QOPT, consulte a secção “Regras de nomes de caminho” na publicação Programação de dispositivos ópticos  .

Conceitos relacionados

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

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 em QOPT.

No entanto os ficheiros em QOPT podem ser acedidos utilizando uma ligação simbólica a partir do sistema de ficheiros “raiz” (/) ou QOpenSys.

Conceitos relacionados

“Ligação” na página 12

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.

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

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QOPT.

A maior parte dos comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 pode operar no sistema de ficheiros QOPT. Existem, contudo, algumas excepções no sistema de ficheiros QOPT. Não se esqueça que pode não ser seguro utilizar estes comandos de CL num processo com capacidade para vários módulos; podem ser aplicáveis determinadas restrições, dependendo do formato do suporte óptico. Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder utilizando menus e ecrãs” na página 72.

Os comandos do sistema de ficheiros integrado que se seguem não são suportados pelo sistema de ficheiros QOPT:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHKIN
- CHKOUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

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

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QOPT.

Todas as APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros “raiz” (/) com protecção por módulos, à excepção das seguintes APIs:

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

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Sistema de ficheiros NetWare (QNetWare)

O sistema de ficheiros QNetWare fornece acesso a dados em servidores de PC autónomos a executar Novell NetWare 5.1 ou 6.0.

Além disso, QNetWare também fornece as seguintes funcionalidades:

- Permite o acesso a objectos de Serviços de Directório de NetWare (NDS).
- Suporta dados armazenados em ficheiros de dados contínuos.
- Permite a montagem dinâmica de sistemas de ficheiros NetWare no espaço de nomes local

Notas:

1. O sistema de ficheiros QNetWare só está disponível quando o programa NetWare Enhanced Integration, opção 25 de BOSS, estiver instalado no sistema. Após o IPL que se segue à instalação, o directório /QNetWare e os respectivos subdirectórios são apresentados como parte da estrutura de directórios do sistema de ficheiros integrado.
2. O produto NetWare Enhanced Integration não suporta NSS (Novell Storage Services), deste modo, o acesso aos dados contidos nessa partição poderá ser limitado ou restrito.

Montar sistemas de ficheiros NetWare

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

A montagem de sistemas de ficheiros Netware também pode ser utilizada para tirar partido das opções do comando Adicionar sistema de ficheiros montado (ADDMFS), como, por exemplo, montar um sistema de ficheiros de leitura e escrita como só de leitura. Consulte o tópico Adicionar sistema de ficheiros montado (ADDMFS) para obter mais informações.

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

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

Estrutura de directórios QNetWare

A estrutura de directórios /QNetWare representa vários sistemas de ficheiros distintos.

- A estrutura representa servidores e volumes Novell NetWare existentes na rede, com o seguinte formato:
/QNetWare/SERVIDOR.SVR/VOLUME
A extensão .SVR é utilizada para representar um servidor Novell NetWare.
- Quando um volume de um servidor é acedido através de menus, comandos ou APIs do sistema de ficheiros integrado, o directório raiz do volume NetWare é automaticamente instalado no directório VOLUME em /QNetWare.
- QNetWare representa árvores NDS na rede com o seguinte formato:
/QNetWare/CORP_TREE.TRE/USA.C/ORG.0/ORG_UNIT.OU/SVR1_VOL.CN
A extensão .TRE é utilizada para representar árvores NDS, .C representa os países, .0 representa organizações, .OU representa unidades organizacionais e .CN é utilizado para representar nomes comuns. Se um volume da Novell Netware for acedido pelo caminho NDS através de um objecto volume ou de um nome alternativo de um objecto volume, o directório raiz também será automaticamente instalado no objecto NDS.

Utilizar o QNetWare através da interface do sistema de ficheiros integrado

É possível aceder ao sistema de ficheiros QNetWare através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros do i5/OS ou as APIs, ecrãs de utilizador e comandos 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 Novell NetWare.

Ao utilizar comandos e APIs para obter ou definir as autoridades de proprietários ou utilizadores, QNetWare mapeia os utilizadores de NetWare para utilizadores do servidor iSeries, com base no nome de um utilizador. Se o nome de NetWare exceder dez caracteres ou se um utilizador do servidor iSeries correspondente não existir, a autoridade não será definida. Os proprietários que não puderem ser definidos serão automaticamente definidos com o perfil do utilizador QDFTOWN. As autoridades dos utilizadores podem ser apresentadas e alteradas utilizando os comandos WRKAUT e CHGAUT. Quando são transferidas de e para o servidor, as autoridades são correlacionadas com as autoridades do servidor iSeries.

Auditoria no sistema de ficheiros QNetWare

Apesar da Novell NetWare suportar a auditoria de ficheiros e de directórios, o sistema de ficheiros QNetWare não pode alterar os valores auditados desses objectos. Deste modo, o comando Alterar Valor de Auditoria (CHGAUD) não é suportado.

Ficheiros e directórios no sistema de ficheiros QNetWare

O sistema de ficheiros QNetWare não mantém as maiúsculas ou minúsculas utilizadas quando os ficheiros e directórios são introduzidos num comando ou numa API.

Todos os nomes são definidos para maiúsculas na transmissão para o servidor NetWare. A Novell NetWare também suporta os espaços de nome de múltiplas plataformas como, por exemplo DOS, OS/2, Apple Macintosh e NFS. O sistema de ficheiros QNetWare suporta apenas o espaço de nome do DOS. Uma vez que o espaço de nome do DOS é necessário em todos os volumes Novell NetWare, todos os ficheiros e directórios aparecerão no sistema de ficheiros QNetWare.

Objectos de NetWare Directory Services no sistema de ficheiros QNetWare

O sistema de ficheiros QNetWare suporta a apresentação de nomes NDS (NetWare Directory Services) 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.

Conceitos relacionados

“Ligação” na página 12

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.

Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QNetWare

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QNetWare

Os comandos incluídos em “Aceder utilizando comandos de CL” na página 74 podem funcionar no sistema de ficheiros QNetWare, com as seguintes excepções:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGJRNOBJ
- CHGPGP
- CHKIN
- CHKOUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

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

- CHGOWN
- CPYFRMSTMF
- CPYTOSTMF
- CRTDIR

Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNetWare

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QNetWare.

As APIs incluídas em “Executar operações utilizando APIs” na página 118 podem funcionar no sistema de ficheiros QNetWare, à excepção das seguintes APIs:

- givedescriptor()
- link()

- QjoEndJournal()
- | • QjoRetrieveJournalEntries()
- 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()

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Sistema de ficheiros do iSeries NetClient (QNTC)

- | O sistema de ficheiros QNTC fornece acesso a dados e objectos armazenados num Servidor xSeries
- | Integrado para iSeries com o Windows NT 4.0 Server ou posterior ou Linux[®]. O sistema de ficheiros
- | QNTC também fornece acesso a dados e objectos armazenados em servidores remotos com o Windows
- | NT 4.0 ou posterior, Linux Samba 3.0 ou posterior, ou versões suportadas do iSeries NetServer[™].

O sistema de ficheiros QNTC permite que as aplicações do iSeries utilizem dados armazenados no Windows ou em servidores do Linux.

O sistema de ficheiros QNTC faz parte do sistema operativo i5/OS base. Não é necessário ter instalado o suporte de servidor integrado, opção 29 do sistema operativo, para aceder ao /QNTC.

Utilizar o QNTC através da interface do sistema de ficheiros integrado

Ao utilizar o iSeries NetServer, o iSeries Navigator, os comandos do sistema de ficheiros integrado, os ecrãs de utilizador ou as APIs pode aceder ao sistema de ficheiros QNTC através da interface do sistema de ficheiros integrado.

Tenha em atenção as 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 do servidor de NT é administrada a partir do servidor do Windows NT. 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 caminho é constituído pelo nome do sistema de ficheiros, pelo nome do servidor, pelo nome da
- | partilha, pelos nomes de directórios e subdirectórios e pelo nome do objecto.

Os requisitos de um nome do caminho são os seguintes:

- O nome de caminho tem de começar por uma barra e pode conter até 255 caracteres. Os nomes de caminho têm o seguinte formato:
`/QNTC/Nomeservidor/Nomepartilha/Directório/ . . . /Objecto`
(QNTC é uma parte obrigatória do nome de caminho.)
- Os nomes de caminho são sensíveis a maiúsculas e minúsculas.
- O nome do servidor pode ter até 15 caracteres. Tem de fazer parte do caminho.
- O nome da partilha pode ter até 12 caracteres.
- Cada componente do nome de caminho, após o nome da partilha, pode ter até 255 caracteres.
- Geralmente, no QNTC, estão disponíveis 130 níveis de hierarquia. Se todos os componentes do nome de caminho estiverem incluídos como níveis de hierarquia, a hierarquia de directórios pode ter até 132 níveis de profundidade.
- Os nomes são armazenados em CCSID Unicode.
- | • Por predefinição, cada servidor funcional suportado na sub-rede local é apresentado automaticamente
- | como um directório em /QNTC. Utilize o comando Criar directório (MKDIR) ou a API mkdir() para
- | adicionar servidores acessíveis exteriores à sub-rede local.

Conceitos relacionados

“Nome de caminho” na página 16

Um *nome de caminho* indica ao servidor como localizar um objecto.

Informações relacionadas

Comando Estabelecer directório (MKDIR)

API Estabelecer directório (mkdir())

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 de QNTC.

Conceitos relacionados

“Ligação” na página 12

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.

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

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QNTC

Os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 podem operar no sistema de ficheiros QNTC, à excepção dos seguintes comandos:

- ADDLNK
- APYJRNCHG
- CHGJRNOBJ
- CHGOWN
- CHGAUT
- CHGPGP
- CHKIN
- CHKOUT
- DSPAUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- RST (disponível com servidores xSeries integrados)
- SAV (disponível com servidores xSeries integrados)
- SNDJRNE
- STRJRN
- WRKAUT
- WRKOBJOWN
- WRKOBJPGP

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas no tópico “Aceder utilizando menus e ecrãs” na página 72.

| Variáveis de ambiente de QNTC

- | O comportamento de procura na rede pode ser controlado através de duas variáveis de ambiente. O suporte para estas variáveis de ambiente iniciou-se no i5/OS V5R4. Utilize o comando de CL
- | ADDENVVAR para criar estas variáveis de ambiente.

| QZLC_SERVERLIST

- | Quando esta variável do ambiente estiver definida como "2", todos os servidores apresentados no directório /QNTC do sistema de ficheiros integrado podem ser acedidos através de QNTC. Este
- | tratava-se do comportamento predefinido anterior à versão V5R4. Quando esta variável não estiver
- | definida como "2" ou não tiver sido criada, alguns servidores apresentados no directório /QNTC poderão
- | não estar acessíveis.

QIBM_ZLC_NO_BROWSE

- | Quando esta variável do ambiente estiver definida como "1", o directório /QNTC conterá apenas os
- | servidores criados utilizando o comando de CL MKDIR ou a API mkdir(). O desempenho de muitas
- | operações no sistema de ficheiros QNTC melhorará quando esta variável do ambiente tiver sido definida.
- | No entanto, nem todos os directórios /QNTC têm de ser criados utilizando o comando de CL.

Criar directórios no sistema de ficheiros QNTC

Utilize o comando Criar directório (MKDIR) ou a API mkdir() para adicionar um directório de servidor ao directório /QNTC.

- | Por predefinição, é criado automaticamente um directório de QNTC para todos os servidores funcionais
- | no domínio do iSeries NetServer e na sub-rede local. Os servidores exteriores à sub-rede local ou ao
- | domínio do iSeries NetServer têm de ser adicionados utilizando o comando MKDIR ou a API mkdir().

| Por exemplo:

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

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

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

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

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

Nota:

- Ao configurar o iSeries NetServer for WINS, é possível criar automaticamente directórios para servidores exteriores à sub-rede.
- Se utilizar a API mkdir() ou o comando de CL MKDIR para adicionar directórios à estrutura de directórios, os mesmos não permanecerão visíveis nos IPLs. O comando MKDIR ou a API mkdir() tem de ser emitido novamente após cada IPL do sistema.

Se pretender adicionar directórios utilizando a API ou o comando de CL, pode melhorar o desempenho destes comandos adicionando a variável do ambiente QIBM_ZLC_NO_BROWSE, conforme exemplificado em seguida:

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

- | Esta variável do ambiente faz com que o sistema de ficheiros ignore todas as procuras na rede ao
- | executar operações de ficheiros.

Informações relacionadas

Comando Estabelecer directório (MKDIR)

API Estabelecer directório (mkdir())

Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QNTC

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QNTC.

As APIs listadas no tópico "Executar operações utilizando APIs" na página 118 podem operar no sistema de ficheiros QNTC, à excepção das seguintes APIs:

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

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

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Activar o sistema de ficheiros para o Serviço de Autenticação da Rede

O QNTC permite o acesso do iSeries aos servidores CIFS que suportem o protocolo de autenticação Kerberos V5.

Em vez de utilizar uma palavra-passe do tipo gestor de rede local para efectuar a autenticação junto de cada servidor, um servidor iSeries devidamente configurado pode agora aceder aos servidores CIFS suportados, através de uma única transacção de início de sessão.

Para activar o Serviço de Autenticação da Rede (NAS - Network Authentication Service) para utilização com o QNTC, tem de configurar os itens:

- Serviço de Autenticação da Rede (NAS)
- Mapeamento de Identidades de Empresa (EIM)

Após a configuração dos itens acima referidos, poderá ser permitido ao utilizador utilizar o NAS com o sistema de ficheiros QNTC. Os passos que se seguem são necessários para permitir que um utilizador tire partido do suporte NAS de QNTC.

- O perfil do utilizador do iSeries tem de ter o parâmetro de gestão de palavra-passe local, LCLPWDMGT, definido como ***NO**. Especificando ***NO**, o utilizador não terá uma palavra-passe de acesso ao sistema e não poderá iniciar sessão numa sessão 5250. O único acesso possível ao sistema será através de aplicações com NAS activado, como o iSeries Navigator ou o iSeries Access 5250 Display Emulator.

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

- Tem de ter uma senha do Kerberos e uma ligação ao iSeries Navigator.
- A senha Kerberos para o iSeries que está a utilizar tem de ser transmissível. Para tornar uma senha transmissível, execute os seguintes passos:
 - Aceda à ferramenta 'Utilizadores e computadores do Active Directory (Active Directory Users and Computers)' em KDC para o nicho NAS
 - Selecciona os utilizadores
 - Selecciona o nome correspondente ao nome principal do serviço
 - Selecciona Propriedades (Properties)

- Seleccione o separador Conta (Account)
- Nas opções de Conta (Account) seleccione 'A conta é fidedigna para delegação'('Account is trusted for delegation').

Informações relacionadas

Serviço de autenticação da rede

Mapeamento de identidade empresarial (EIM)

Sistema de ficheiros do servidor de ficheiros do i5/OS (QFileSvr.400)

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

O sistema de ficheiros QFileSvr.400 pode ser considerado como um cliente que actua em nome dos utilizadores para executar pedidos de ficheiros. O QFileSvr.400 interage com o servidor de ficheiros do i5/OS no sistema destino para executar a operação de ficheiro real.

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

É possível aceder ao sistema de ficheiros QFileSvr.400 através da interface do sistema de ficheiros integrado utilizando o servidor de ficheiros do i5/OS ou as APIs, ecrãs de utilizador e comandos do sistema de ficheiros integrado.

Ao utilizar as interfaces do sistema de ficheiros integrado, deverá ter em conta 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 QFileSvr.400

Num directório de primeiro nível, que na verdade 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 efectuada 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 i5/OS.

Nomes de caminhos no sistema de ficheiros QFileSvr.400

Os nomes de caminhos têm um formato específico no sistema de ficheiros QFileSvr.400.

- O formato é o seguinte:

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

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

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

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

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

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

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

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

Conceitos relacionados

“Continuidade de nomes” na página 19

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Comunicações no sistema de ficheiros QFileSvr.400

O sistema de ficheiros QFileSvr.400 comunica da forma seguinte.

- 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). Ao estabelecer 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 Programação de APPC  .

- 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 utilizem SNA como protocolo de comunicações são executados pelo trabalho Q400FILSVR do sistema i5/OS.
- Se ainda não tiver sido estabelecida ligação ao servidor destino, o sistema de ficheiros QFileSvr.400 assume que o directório de primeiro nível representa um nome de sistema central de TCP/IP. Para estabelecer uma ligação ao servidor destino, o sistema de ficheiros QFileSvr.400 executa os seguintes passos:
 1. Atribui o nome da localização remota a um endereço de IP.
 2. Liga ao programa de definição de correspondências do servidor do sistema central na porta 449 já conhecida utilizando o endereço de IP resolvido. Em seguida, envia ao programa de definição de correspondências do servidor uma consulta do nome do serviço “como-ficheiro.” Uma das seguintes situações ocorre como resultado da consulta:
 - Se “como-ficheiro” estiver na tabela de serviços do servidor destino, o programa de mapeamento do servidor devolve a porta à qual o daemon do servidor de ficheiros do i5/OS está a escutar.
 - Se o programa de definição de correspondências do servidor não estiver activo no servidor destino, será utilizado o número de porta predefinido de “como-ficheiro” (8473).

Em seguida, o sistema de ficheiros QFileSvr.400 tenta estabelecer uma ligação TCP ao daemon do servidor de ficheiros do i5/OS no servidor destino. Quando a ligação for estabelecida, o QFileSvr.400 troca pedidos e respostas com o servidor de ficheiros. No subsistema QSERVER, os pedidos iniciados previamente QPWFSERVSO têm o controlo da ligação. Cada trabalho iniciado previamente é executado sob o respectivo perfil de utilizador.

3. Se o nome da localização remota não for resolvido como um endereço de IP, o directório de primeiro nível é considerado como um nome de LU 6.2 de SNA. Assim, é feita uma tentativa para estabelecer uma ligação APPC com o servidor de ficheiros do i5/OS.
- 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 qualquer 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 consegue 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 sobre o objecto no sistema de ficheiros QFileSvr.400

- | Se ambos os sistemas tiverem o Serviço de Autenticação da Rede e o Mapeamento de Identidades de Empresa (EIM - Enterprise Identity Mapping) configurados e o utilizador tiver efectuado a autenticação utilizando Kerberos, este sistema também pode ser utilizado para efectuar a autenticação de acesso ao sistema de ficheiros que reside num servidor iSeries destino.

Se a autenticação Kerberos falhar, o ID de utilizador e a palavra-passe podem ser utilizados para verificar o acesso.

Nota: Se a senha de concessão de senhas ou a senha do servidor expirar após o servidor destino ter verificado o seu acesso, a expiração só será efectiva quando a ligação ao servidor destino tiver terminado.

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

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

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

Informações relacionadas

Serviço de autenticação da rede

Mapeamento de identidade empresarial (EIM)

Ligações no sistema de ficheiros QFileSvr.400

O sistema de ficheiros QFileSvr.400 suporta apenas uma ligação a um objecto.

Não é possível criar nem armazenar ligações simbólicas em QFileSvr.400. No entanto os ficheiros em QFileSvr.400 podem ser acedidos utilizando uma ligação simbólica a partir dos sistemas de ficheiros "raiz" (/), QOpenSys ou definidos pelo utilizador.

Conceitos relacionados

“Ligação” na página 12

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.

Utilizar comandos e ecrãs do sistema de ficheiros integrado no sistema de ficheiros QFileSvr.400

Existem vários comandos e ecrãs do sistema de ficheiros integrado que são válidos no sistema de ficheiros QFileSvr.400.

Os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 podem operar no sistema de ficheiros QFileSvr.400, à excepção dos seguintes comandos:

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

Aplicam-se as mesmas restrições aos ecrãs do utilizador, que as descritas em “Aceder utilizando menus e ecrãs” na página 72.

Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros QFileSvr.400

Existem várias APIs do sistema de ficheiros integrado que são válidas no sistema de ficheiros QFileSvr.400.

As APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros QFileSvr.400, à excepção das seguintes APIs:

- chown()
- fchown()
- fclear()
- fclear64()
- givedescriptor()
- link()
- QjoEndJournal()
- | • QjoRetrieveJournalEntries()
- QjoRetrieveJournalInformation()

- QJorjidi()
- QJOSJRNE
- QjoStartJournal
- Qp0lGetPathFromFileID()
- symlink()
- takedescriptor()

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Network File System (NFS)

O NFS (Network File System) permite ao utilizador aceder a dados e objectos armazenados num servidor de NFS remoto.

Um servidor de NFS pode exportar um Network File System que posteriormente será instalado dinamicamente pelos clientes de NFS.

Além disso, qualquer sistema de ficheiros instalado localmente através do Network File System 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.

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

É possível aceder ao NFS através da interface do sistema de ficheiros integrado. Tenha em atenção estas considerações e limitações.

Características do Network File System

As características de qualquer sistema de ficheiros montado através de NFS dependem do tipo de sistema de ficheiros montado a partir do servidor.

É importante compreender que os pedidos executados no que parece ser um directório ou sistema de ficheiros local estão, de facto, a ser executados no servidor, através da ligação NFS.

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

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

Variações de servidores e clientes no Network File System

Existem três grandes possibilidades para as ligações de cliente/servidor que podem afectar o modo como o Network File System (NFS) funcionará, bem como as respectivas características.

As três possibilidades são:

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

3. O utilizador monta um sistema de ficheiros num cliente a partir de um servidor que não seja um servidor iSeries nem um servidor UNIX.

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

No segundo cenário, é importante perceber que qualquer sistema de ficheiros instalado a partir de um servidor UNIX terá um comportamento muito semelhante ao do sistema de ficheiros QOpenSys do servidor iSeries.

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

Referências relacionadas

“Sistema de ficheiros de sistemas abertos (QOpenSys)” na página 36

O sistema de ficheiros QOpenSys é compatível com as normas do sistema aberto com base em UNIX, como POSIX e X/Open Portability Guide (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.

Ligações no Network File System

Em geral, são permitidas múltiplas ligações permanentes ao mesmo objecto no Network File System.

As ligações simbólicas são completamente suportadas. Pode ser utilizada uma ligação simbólica para ligar um Network File System a um objecto de outro sistema de ficheiros. As capacidades de múltiplas ligações permanentes e simbólicas dependem por completo do sistema de ficheiros que estiver a ser instalado com o NFS.

Conceitos relacionados

“Ligação” na página 12

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.

Comandos do sistema de ficheiros integrado no Network File System

Existem vários comandos do sistema de ficheiros integrado que são válidas no Network File System (NFS).

Todos os comandos listados no tópico “Aceder utilizando comandos de CL” na página 74 e os ecrãs descritos no tópico “Aceder utilizando menus e ecrãs” na página 72 podem operar no Network File System, à excepção dos seguintes comandos:

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

- CHKOUT
- | • DSPJRN
- ENDJRN
- | • RCLLNK
- | • RCVJRNE
- | • RTVJRNE
- SNDJRNE
- STRJRN

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

Network File System no OS/400. 

Tabela 6. Comandos de CL do Network File System

Comando	Descrição
ADDMFS	Adicionar sistema de ficheiros instalado. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais.
CHGNFSEXP	Alterar Network File System. Exporta, adiciona ou remove árvores de directórios para a tabela de exportações de sistemas de ficheiros exportados para clientes do Network File System.
DSPMFSINF	Visualizar informações sobre o sistema de ficheiros montado. Apresenta informações sobre um sistema de ficheiros montado.
ENDNFSSVR	Terminar servidor do sistema de ficheiros da rede. Termina um ou todos os daemons do Network File System no servidor.
EXPORTFS	Exportar um sistema de ficheiros. Adiciona ou remove árvores de directórios para a tabela de exportações de sistemas de ficheiros exportados para clientes do Network File System.
MOUNT	Instalar um sistema de ficheiros. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais. Este comando é um nome alternativo correspondente ao comando ADDMFS.
RLSIFSLCK	Libertar bloqueios do sistema de ficheiros integrado. Liberta todos os bloqueios no intervalo de bytes do Network File System suportados por um cliente ou de um objecto.
RMVMFS	Remover sistema de ficheiros instalado. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local.
STRNFSSVR	Iniciar servidor do sistema de ficheiros da rede. Inicia um ou todos os daemons do Network File System no servidor.
UNMOUNT	Desinstalar um sistema de ficheiros. Remove os sistemas de ficheiros exportados do servidor remoto do espaço de nome do cliente local. Este comando é um nome alternativo correspondente ao comando RMVMFS.

Nota: É necessário instalar um Network File System para que seja possível utilizar comandos no mesmo.

APIs do sistema de ficheiros integrado no Network File System

Existem várias APIs do sistema de ficheiros integrado que são válidas no Network File System (NFS).

Todas as APIs listadas no tópico “Executar operações utilizando APIs” na página 118 podem operar no sistema de ficheiros da rede, à excepção das seguintes APIs:

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

Para obter uma descrição completa das funções de linguagem C directamente relacionadas com o Network File System, consulte o manual Suporte de Network File System no OS/400 .

Nota: É necessário instalar um Network File System para que seja possível utilizar APIs no mesmo.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Aceder ao sistema de ficheiros integrado

Todas as interfaces de utilizador, tais como menus, comandos e ecrãs, utilizadas para trabalhar com bibliotecas, objectos, ficheiros de base de dados, pastas e documentos do sistema continuam a funcionar tal como funcionavam antes da introdução do sistema de ficheiros integrado.

No entanto, estas interfaces não podem ser utilizadas para trabalhar com ficheiros de dados contínuos, directórios nem com outros objectos suportados pelo sistema de ficheiros integrado.

É fornecido um conjunto de interfaces de utilizador em separado 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, através da utilização de menus e ecrãs ou de comandos da linguagem de controlo (CL). Além disso, pode utilizar interfaces de programas de aplicações (APIs) para tirar partido dos ficheiros de dados contínuos, directórios e outro suporte do sistema de ficheiros integrado.

Também pode interagir com o sistema de ficheiros integrado através do iSeries Navigator, uma interface gráfica de utilizador (GUI - Graphical User Interface) utilizada para gerir e administrar o servidor a partir da área de trabalho do Windows.

Aceder utilizando menus e ecrãs

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

Para apresentar menus do sistema de ficheiros integrado:

1. Inicie sessão no servidor.
2. Prima Enter para continuar.
3. No menu principal do iSeries, seleccione a opção **Ficheiros, bibliotecas e pastas (Files, Libraries, and Folders)**.
4. No menu Ficheiros, bibliotecas e pastas (Files, Libraries, and Folders), seleccione a opção **Sistema de ficheiros integrado (Integrated File System)**.

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

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

1. Escreva GO DATA em qualquer linha de comandos para visualizar o menu Ficheiros, Bibliotecas e Pastas.
2. Selecciona **Sistema de ficheiros integrado (Integrated file system)**.

Para ver um menu de comandos do Network File System, escreva GO CMDNFS em qualquer linha de comandos. Para ver um menu de comandos do sistema de ficheiros definido pelo utilizador, escreva GO CMDUDFS em qualquer linha de comandos.

Nos menus do sistema de ficheiros integrado pode solicitar ecrãs ou comandos que permitam efectuar as seguintes operações:

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

Alguns sistemas de ficheiros não suportam todas estas operações.

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Referências relacionadas

“Regras de nomes de caminhos para comandos de CL e ecrãs” na página 77

Ao utilizar um comando ou ecrã do sistema de ficheiros integrado para operar um objecto, este é identificado fornecendo o respectivo nome de caminho.

“Aceder utilizando comandos de CL” na página 74

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros acessível através da interface do sistema de ficheiros integrado.

“Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros “raiz” (/)” na página 35

Todos os comandos listados no tópico “Aceder utilizando comandos de CL” e os ecrãs descritos no tópico “Aceder utilizando menus e ecrãs” podem operar no sistema de ficheiros “raiz” (/). No entanto, a utilização destes comandos num processo que suporte vários módulos pode não oferecer segurança.

Aceder utilizando comandos de CL

Todas as operações possíveis através dos menus e ecrãs do sistema de ficheiros integrado podem ser efectuadas introduzindo comandos de CL (Control Language). Estes comandos podem ser operados em ficheiros e outros objectos de qualquer sistema de ficheiros 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, sobre o Network File System e sistemas de ficheiros montados, em geral, consulte os tópicos "Sistemas de ficheiros definidos pelo utilizador (UDFSs)" na página 38 e "Network File System (NFS)" na página 69. Sempre que um comando execute a mesma operação que um comando do OS/2 ou do DOS, é fornecido um nome de alternativo (como nome de comando alternativo) para comodidade dos utilizadores do OS/2 e do DOS.

Tabela 7. Comandos do sistema de ficheiros integrado

Comando	Descrição	Nome alternativo
ADDLNK	Adicionar ligação. Adiciona uma ligação entre um directório e um objecto.	
ADDMFS	Adicionar sistema de ficheiros instalado. Coloca sistemas de ficheiros exportados do servidor remoto em directórios de cliente locais.	MOUNT
APYJRNCHG ²	Aplicar alterações registadas em diário. Utiliza entradas de diário para aplicar alterações ocorridas desde que um objecto registado em diário foi guardado ou para aplicar alterações até um determinado ponto.	
CHGATR	Alterar atributo. Altera um atributo para um único objecto, para um grupo de objectos ou para uma árvore de directórios onde o directório, o respectivo conteúdo e o conteúdo de todos os subdirectórios nele incluídos que têm o atributo alterado.	
CHGAUD	Alterar valor de auditoria. Activa ou desactiva a auditoria de um objecto.	
CHGAUT	Alterar autoridade. Concede uma autoridade específica para um objecto a um utilizador ou a um grupo de utilizadores.	
CHGCURDIR	Alterar directório actual. Altera o directório para ser utilizado como directório actual.	CD, CHDIR
CHGJRNOBJ ²	Alterar objectos registados em diário. Altera os atributos de registo em diário de um objecto ou lista de objectos sem ter de terminar e reiniciar o registo em diário do objecto.	
CHGNFSEXP	Alterar exportação do sistema de ficheiros da rede. Adiciona árvores de directórios ou remove-as da tabela de exportação que é exportada para os clientes de NFS.	EXPORTFS
CHGOWN	Alterar proprietário. Transfere a propriedade de um objecto de um utilizador para outro.	
CHGPGP	Alterar grupo principal. Altera o grupo principal de um utilizador para outro.	
CHKIN	Dar entrada. Dá entrada a um objecto ao qual previamente tinha sido dada saída.	
CHKOUT	Dar saída. Dá saída a um objecto, o que impede que seja alterado por outros utilizadores.	
CPY	Copiar. Copia um único objecto ou um grupo de objectos.	COPY

Tabela 7. Comandos do sistema de ficheiros integrado (continuação)

Comando	Descrição	Nome alternativo
CPYFRMSTMF	Copiar de ficheiro de dados contínuos. Copia dados de um ficheiro de dados contínuos para um membro de ficheiro de base de dados.	
CPYTOSTMF	Copiar para ficheiro de dados contínuos. Copia dados de um membro de ficheiro de base de dados para um ficheiro de dados contínuos.	
CRTDIR	Criar directório. Adiciona um novo directório ao sistema.	MD, MKDIR
CRTUDFS	Criar UDFS. Cria um Sistema de Ficheiros Definido pelo Utilizador.	
CVTDIR	Converter directório. Fornece informações acerca da conversão de directórios do sistema de ficheiros integrado do formato *TYPE1 para o formato *TYPE2.	
CVTRPCSRC	Converter origem de RPC. Gera código C de um ficheiro de entrada escrito na linguagem de Chamada de Procedimento Remoto (RPC).	RPCGEN
DLTUDFS	Eliminar UDFS. Elimina um Ficheiro Definido pelo Utilizador.	
DSPAUT	Visualizar autoridade. Mostra uma lista de utilizadores autorizados de um objecto e as respectivas autoridades para o mesmo.	
DSPCURDIR	Visualizar directório actual. Mostra o nome do directório actual.	
DSPJRN ²	Visualizar diário. Converte as entradas de diário (existentes em um ou mais receptores) num formato adequado à representação externa.	
DSPLNK	Visualizar ligações de objecto. Mostra uma lista de objectos num directório e fornece opções de visualização das informações referentes aos objectos.	
DSPF	Visualizar ficheiro de dados contínuos. Apresenta um ficheiro de dados contínuos ou um ficheiro de base de dados.	
DSPMFSINF	Visualizar informações sobre o sistema de ficheiros montado. Apresenta informações sobre um sistema de ficheiros montado.	STATFS
DSPUDFS	Visualizar UDFS. Mostra o Sistema de Ficheiros Definido pelo Utilizador.	
EDTF	Editar ficheiro de dados contínuos. Edita um ficheiro de dados contínuos ou um ficheiro de base de dados.	
ENDJRN ²	Terminar registo em diário. Termina o registo em diário das alterações efectuadas a um objecto ou lista de objectos.	
ENDNFSSVR	Terminar servidor do sistema de ficheiros da rede. Termina um ou todos os daemons de NFS no servidor e no cliente.	
ENDRPCBIND	Terminar daemon do processador de enlacede RPC. Termina o daemon RPCBind de Chamada de Procedimento Remoto (RPC).	
MOV	Mover. Move um objecto para um directório diferente.	MOVE
PRDIRINF	Imprimir informações do directório. É utilizado para imprimir informações do directório relativas a objectos do sistema de ficheiros integrado recolhidas através do comando Obter informações do directório (RTVDIRINF).	
RCLLNK	Regenerar ligações de objectos. Identifica e, se possível, corrige problemas em sistemas de ficheiros montados em utilização.	

Tabela 7. Comandos do sistema de ficheiros integrado (continuação)

Comando	Descrição	Nome alternativo
RCVJRNE ²	Receber entrada de diário. Permite a um programa de saída do utilizador especificado receber de forma contínua entradas de diário.	
RLSIFSLCK	Libertar bloqueios do sistema de ficheiros integrado. Liberta todos os bloqueios no intervalo de bytes retidos por um cliente de NFS ou de um objecto.	
RMVDIR	Remover directório. Remove um directório do sistema.	RD, RMDIR
RMVLNK	Remover ligação. Remove a ligação a um objecto.	DEL, ERASE
RMVMFS	Remover sistema de ficheiros instalado. Remove os sistemas de ficheiros exportados do servidor remoto dos directórios de cliente locais.	UNMOUNT
RNM	Mudar o nome. Altera o nome de um objecto num directório.	REN
RPCBIND	Iniciar daemon de processador de enlaces de RPC. Inicia o Daemon RPCBind de chamada de procedimento remoto (RPC).	
RST	Restaurar. Copia um objecto ou grupo de objectos de um dispositivo de cópia de segurança para o sistema.	
RTVCURDIR	Obter directório actual. Obtém o nome do directório actual e coloca-o numa variável especificada (utilizada em programas de CL)	
RTVDIRINF	Obter informações do directório. É utilizado para recolher atributos de objectos no sistema de ficheiros integrado.	
RTVJRNE ²	Obter entrada de diário. Permite obter uma entrada de diário específica e apresentar os resultados em variáveis de CL.	
SAV	Guardar. Copia um objecto ou grupo de objectos do sistema para um dispositivo de cópia de segurança.	
SNDJRNE ²	Enviar entrada de diário. Adiciona as entradas de diário do utilizador, associadas por opção a um objecto registado em diário, a um receptor de diário.	
STRJRN ²	Iniciar registos em diário. Inicia o registo de alterações em diário (efectuadas num objecto ou numa lista de objectos) num diário específico.	
STRNFSSVR	Iniciar servidor do sistema de ficheiros da rede. Inicia um ou todos os daemons de NFS no servidor e no cliente.	
WRKAUT	Trabalhar com autoridade. Mostra uma lista de utilizadores e as respectivas autoridades e fornece opções para adicionar um utilizador, alterar uma autoridade de utilizador ou remover um utilizador.	
WRKLNK	Trabalhar com ligações de objectos. Mostra uma lista de objectos num directório e fornece opções para executar acções nos objectos.	
WRKOBJOWN ¹	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 ¹	Trabalhar com objectos por grupo principal. Mostra uma lista de objectos controlados por um grupo principal e fornece opções para executar acções nos objectos.	

Notas:

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.

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Tarefas relacionadas

“Aceder utilizando menus e ecrãs” na página 72

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

Referências relacionadas

“Utilizar comandos do sistema de ficheiros integrado no sistema de ficheiros “raiz” (/)” na página 35
Todos os comandos listados no tópico “Aceder utilizando comandos de CL” e os ecrãs descritos no tópico “Aceder utilizando menus e ecrãs” podem operar no sistema de ficheiros “raiz” (/). No entanto, a utilização destes comandos num processo que suporte vários módulos pode não oferecer segurança.

Informações relacionadas

CL (Control language)

Regras de nomes de caminhos para comandos de CL e ecrãs

Ao utilizar um comando ou ecrã do sistema de ficheiros integrado para operar um objecto, este é identificado fornecendo o respectivo nome de caminho.

A lista que se segue constitui um resumo das regras a ter em consideração ao especificar nomes de caminho. Nestas regras, o termo *objecto* refere-se a qualquer directório, ficheiro, ligação ou outro objecto:

- Os nomes dos objectos devem ser exclusivos dentro de cada directório.
- O nome de caminho que é transferido para um comando de CL do sistema de ficheiros integrado tem de ser representado no CCSID presentemente em vigor para o trabalho. Se o CCSID do trabalho for 65535, o nome de caminho tem de ser representado no CCSID predefinido 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 colocados 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 operado pelo comando. O nome de cada componente no caminho é separado por uma barra (/).

Nota: Alguns comandos de CL também permitem que a barra invertida (\) seja utilizada como separador convertendo automaticamente a barra invertida (\) numa barra (/). No entanto, outros comandos de CL, não têm a barra invertida (\) em consideração de forma diferente de qualquer outro carácter. Deste modo, o separador barra invertida (\) deverá ser utilizado com cuidado.

Por exemplo:

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

ou

```
'Dir1\Dir2\Dir3\UsrFile'
```

- Os caracteres barra (/) e barra invertida (\) e os nulos não podem ser utilizados nos componentes individuais do nome de caminho quando a barra (/) e a barra invertida (\) são utilizadas como separadores. As minúsculas não são alteradas para maiúsculas pelos comandos. O nome pode ou não

ser alterado para maiúsculas, dependendo do facto de o sistema de ficheiros que contém o objecto ser sensível a maiúsculas e minúsculas e de o objecto estar a ser criado ou procurado.

- O comprimento do nome de um objecto está limitado pelo sistema de ficheiros onde o objecto se encontra e pelo comprimento máximo de uma cadeia de comandos. Os comandos aceitam nomes de objecto até 255 caracteres de comprimento e nomes de caminho até 5000 caracteres.
- Um carácter separador (por exemplo: /) no início de um nome de caminho significa que o caminho começa no directório mais acima, directório “raiz” (/); por exemplo:

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

- Se o nome do caminho não começar por um carácter separador (por exemplo: /), pressupõe-se que o caminho comece 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 carácter til (~) seguido de um carácter separador (por exemplo: /) 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 carácter til (~) seguido de um nome do utilizador e, em seguida, de um carácter separador (por exemplo: /) 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 comecem por *d* seguido 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 operar 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'
```

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

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

- O nome do caminho têm de estar colocado entre plicas (') ou aspas (") adicionais se qualquer dos caracteres que se seguem for utilizado num nome de componente:

– Asterisco (*)

Nota: Para evitar uma possível confusão com valores especiais do servidor iSeries, os nomes de caminhos não devem ser iniciados por um único carácter de 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 de forma incorrecta.

- 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 códigos forem utilizados, têm de ser introduzidos como uma representação hexadecimal, conforme exemplificado em seguida:

```
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 aplica-se apenas a comandos e ecrãs associados e não a APIs. Além disso, não é permitido um valor hexadecimal 0 em nomes de caminho.

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

“Nome de caminho” na página 16

Um *nome de caminho* indica ao servidor como localizar um objecto.

Referências relacionadas

“Regras de nomes de caminhos para APIs” na página 126

Quando utilizar uma API de ILE C/400® ou do sistema de ficheiros integrado para operar um objecto, o objecto é identificado fornecendo o respectivo caminho do directório. Segue-se um resumo das regras a ter em consideração ao especificar nomes de caminhos nas APIs.

Informações relacionadas

CL (Control language)

Trabalhar com a saída dos comandos RTVDIRINF e PRTDIRINF

O comando Obter informações do directório (RTVDIRINF) é utilizado para recolher atributos para objectos do sistema de ficheiros integrado. As informações recolhidas são armazenadas em ficheiros de base de dados (tabelas) que são designados através do prefixo do ficheiro das informações especificado no parâmetro INFFILEPFX. As tabelas são criadas na biblioteca especificada pelo parâmetro INFLIB.

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

A tabela que se segue descreve os campos fornecidos para a tabela que armazena os atributos dos objectos. Se especificar *GEN no parâmetro de prefixo do ficheiro de informações (INFFILEPFX), os ficheiros da base de dados são criados com um prefixo único gerado por este comando. O prefixo começa por QAEZD seguido de quatro dígitos. Os ficheiros criados para armazenar as informações recolhidas são denominados utilizando este prefixo seguido da letra D (para o ficheiro que contém as informações do directório) ou da letra O (para o ficheiro que contém as informações sobre os objectos nos directórios). Por exemplo, quando o comando for executado pela primeira vez e *GEN tiver sido especificado, os ficheiros QAEZD0001D e QAEZD0001O são criados na biblioteca especificada pelo parâmetro Biblioteca

de Informações (INFLIB). Os utilizadores podem especificar um prefixo do ficheiro até nove caracteres para ser utilizado na designação desta base de dados.

Tabela 8. QAEZDxxxxO (armazenar atributos de objectos)

Nome do campo	Tipo de campo	Descrição do campo
QEZACCTIM	TIMESTAMP	A data e a hora do último acesso aos dados do objecto.
QEZALCSIZE ¹	BIGINT	O número de bytes atribuídos a este objecto.
QEZALWCKPW	SMALLINT	Em que casos um ficheiro de dados contínuos(*STMF) pode ser partilhado com leitores e escritores durante o processamento de ponto de controlo guardar-enquanto-activo. Os valores válidos são: 0 - O objecto só pode ser partilhado com leitores. 1 - O objecto pode ser partilhado com leitores e escritores.
QEZASP	SMALLINT	O conjunto de memória auxiliar no qual está armazenado o objecto.
QEZAUDT	GRAPHIC (10)	O valor de auditoria associado ao objecto. Os valores válidos são: *NONE - Não é efectuada uma auditoria a este objecto se o mesmo for lido ou alterado independentemente do utilizador que está a aceder ao objecto. *USRPRF - Só efectuar uma auditoria a este objecto se estiver a ser efectuada uma auditoria ao utilizador actual. É efectuado um teste ao utilizador actual para determinar se deve ser efectuada uma auditoria a este objecto. É possível especificar através do perfil do utilizador se a auditoria é efectuada apenas ao acesso para alterações ou aos acessos para leitura e alterações a este objecto. *CHANGE - Efectuar uma auditoria a todos os acessos para alterações a este objecto por todos os utilizadores do sistema. *ALL - Efectuar uma auditoria a todos os acessos a este objecto por todos os utilizadores do sistema. Todos os acessos são definidos como uma operação de leitura ou alteração.
QEZAUTLST	GRAPHIC (10)	O nome da lista de autorizações utilizada para proteger o objecto designado. O valor *NONE indica que não está a ser utilizada uma lista de autorizações para determinar a autoridade do objecto.
QEZBLKSIZ	INTEGER	O tamanho de bloco de um objecto.
QEZCASE	SMALLINT	Indica se o sistema de ficheiros que contém este objecto é sensível a maiúsculas e minúsculas. 0 - O sistema de ficheiros não é sensível a maiúsculas e minúsculas. 1 - O sistema de ficheiros é sensível a maiúsculas e minúsculas.
QEZCCSID	INTEGER	O CCSID dos dados e atributos expandidos do objecto.
QEZCEAS	BIGINT	Número de atributos expandidos críticos associados a este objecto.
QEZCHGTIMA ¹	TIMESTAMP	A data e a hora da última modificação dos atributos do objecto.
QEZCHGTIMD	TIMESTAMP	A data e a hora da última modificação dos dados do objecto.
QEZCHKOUT ¹	SMALLINT	Um indicador sempre que for dada saída a um objecto. Os valores válidos são: 0 - Não foi dada saída ao objecto. 1 - Foi dada saída ao objecto.
QEZCHKOWN	GRAPHIC (10)	O utilizador que deu saída ao objecto. Se não foi dada saída ao objecto, este campo estará em branco.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZCHKTIM	TIMESTAMP	A data e a hora de saída do objecto. Se não tiver sido dada saída do objecto, este campo terá como valor NULL.
QEZCLSTRSP	SMALLINT	O objecto corresponde ao armazenamento alocado aos servidores xSeries integrados a utilizar como unidades de disco virtuais para os servidores xSeries. Na perspectiva do servidor iSeries, as unidades virtuais aparecem como ficheiros de dados contínuos em bytes dentro do sistema de ficheiros integrado. 0 - O objecto não está na memória em disco virtual. 1 - O objecto está na memória em disco virtual.
QEZCRTAUD	GRAPHIC (10)	O valor de auditoria associado a um objecto criado neste directório. Os valores válidos são: *NONE - Não é efectuada uma auditoria a este objecto se o mesmo for lido ou alterado independentemente do utilizador que está a aceder ao objecto. *USRPRF - Só efectuar uma auditoria a este objecto se estiver a ser efectuada uma auditoria ao utilizador actual. É efectuado um teste ao utilizador actual para determinar se deve ser efectuada uma auditoria a este objecto. É possível especificar através do perfil do utilizador se a auditoria é efectuada apenas aos acessos para alterações ou aos acessos para leitura e alterações a este objecto. *CHANGE - Efectuar uma auditoria a todos os acessos para alterações a este objecto por todos os utilizadores do sistema. *ALL - Efectuar uma auditoria a todos os acessos a este objecto por todos os utilizadores do sistema. Todos os acessos são definidos como uma operação de leitura ou alteração.
QEZCRTTIM	TIMESTAMP	A data e a hora de criação do objecto.
QEZDIRIDX	INTEGER	O índice remissivo do directório ascendente.
QEZDIRTYP2	SMALLINT	O formato do objecto do directório especificado. Os valores válidos são: 0 - O formato do directório é *TYPE1. 1 - O formato do directório é *TYPE2.
QEZDOM	GRAPHIC (10)	O domínio do objecto. Os valores válidos são: *SYSTEM - O objecto existe no domínio do sistema. *USER - O objecto existe no domínio do utilizador.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZDSTGOPT	SMALLINT	<p>Esta opção deverá ser utilizada para determinar a atribuição de memória auxiliar pelo sistema ao objecto especificado. Esta opção só poderá ser especificada para ficheiros de dados contínuos nos sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador. Esta opção será ignorada nos ficheiros de dados contínuos de bytes *TYPE1. Os valores válidos são:</p> <p>0 - A memória auxiliar será atribuída normalmente. Ou seja, uma vez que é necessário uma memória auxiliar adicional, esta será atribuída num tamanho lógico de forma a satisfazer os requisitos de espaço actuais e futuros, ao mesmo tempo que reduz o número de operações de I/O efectuadas no disco.</p> <p>1 - A memória auxiliar será atribuída para reduzir o espaço utilizado pelo objecto. Ou seja, uma vez que é necessário uma memória auxiliar adicional, esta será atribuída com dimensões reduzidas de forma a satisfazer os requisitos de espaço actuais. O acesso a um objecto composto por muitas extensões pequenas poderá aumentar o número de operações de I/O do disco relativas a esse objecto.</p> <p>2 - O sistema determinará de forma dinâmica a atribuição de memória auxiliar ideal para o objecto, o equilíbrio de espaço utilizado versus as operações de I/O do disco. Por exemplo, se um ficheiro tiver muitas extensões pequenas, for lido e escrito com frequência, futuramente será atribuída mais memória auxiliar para reduzir o número de operações de I/O do disco. Ou, se um ficheiro for truncado com frequência, futuramente será atribuída menos memória auxiliar para reduzir o espaço utilizado. Além disso, as informações serão mantidas nos tamanhos dos ficheiros de dados contínuos do sistema e das respectivas actividades. Estas informações de tamanho dos ficheiros também serão utilizadas para ajudar a determinar a atribuição de memória auxiliar ideal para este objecto uma vez que está relacionado com outros tamanhos de objectos.</p>
QEZDTASIZE	BIGINT	O tamanho em bytes dos dados deste objecto. Este tamanho não inclui os cabeçalhos do objecto ou o tamanho dos atributos expandidos associados ao objecto.
QEZEAS	BIGINT	Número de atributos expandidos associados a este objecto.
QEZEXTATRS	BIGINT	Número total de bytes de todos os dados de atributos expandidos.
QEZFILEID ¹	GRAPHIC (16)	O ID de ficheiro do objecto. Um identificador associado ao objecto. É possível utilizar um ID de ficheiro com o Qp0lGetPathFromFileID() para obter um nome de caminho do objecto.
QEZFILEIDS	INTEGER	O ID de ficheiro de 4 bytes do ficheiro. Este número identifica unicamente o objecto num sistema de ficheiros. Não é possível através deste número identificar o objecto em todo o sistema.
QEZFILYP2 ¹	SMALLINT	<p>O formato de ficheiro de dados contínuos (*STMF). Os valores válidos são:</p> <p>0 - O formato de ficheiro de dados contínuos é *TYPE1.</p> <p>1 - O formato de ficheiro de dados contínuos é *TYPE2.</p>
QEZFSID	BIGINT	O ID do sistema de ficheiros a que pertence o objecto. Este número identifica unicamente o sistema de ficheiros a que pertence o objecto.
QEZGENID	BIGINT	O ID de geração associado ao ID de ficheiro.
QEZGID	INTEGER	A identificação dos perfis de grupo é efectuada através de um número de identificação de grupo (GID) numérico exclusivo.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZINHSCN	GRAPHIC (1)	<p>Se os objectos criados num directório serão pesquisados quando os programas de saída são registados através de um dos pontos de saída relacionados com a pesquisa do sistema de ficheiros integrado.</p> <p>Os valores válidos são:</p> <p>x'00' - Após ter sido criado um objecto no directório, o objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa.</p> <p>Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p> <p>x'01' - Após um objecto ter sido criado no directório, o objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.</p> <p>x'02' - Após um objecto ter sido criado no directório, o objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa apenas se o objecto tiver sido modificado desde a última pesquisa efectuada ao objecto. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado. Este atributo só entrará em vigor se o valor do sistema Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) tiver *USEOCOATR especificado. De outro modo, será considerado como se o atributo fosse SCANNING_YES.</p> <p>Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p>
QEZJAFTERI	SMALLINT	<p>Se o registo em diário estiver activo, a imagem do objecto após uma alteração é registada.</p> <p>0 - O registo em diário do objecto não é efectuado com imagens posteriores.</p> <p>1 - O registo em diário do diário é efectuado com imagens posteriores.</p>
QEZJBEFORI	SMALLINT	<p>Se o registo em diário estiver activo, a imagem do objecto é registada no diário antes de uma alteração.</p> <p>0 - O registo em diário do objecto não é efectuado com imagens anteriores.</p> <p>1 - O registo em diário do objecto é efectuado com imagens anteriores.</p>
QEZJOPTENT	SMALLINT	<p>Se o registo em diário estiver activo é efectuado o registo das entradas consideradas opcionais. A lista de entradas de diário opcionais varia consoante o tipo de objecto.</p> <p>0 - O registo em diário do objecto não é efectuado com as entradas opcionais.</p> <p>1 - O registo em diário do objecto é efectuado com as entradas opcionais.</p>

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

Nome do campo	Tipo de campo	Descrição do campo
QEZJRCVASP	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações do diário ao nome do ASP que contém o receptor de diário. Os valores válidos são: *SYSBAS - O receptor de diário encontra-se no sistema ou no ASP do utilizador. Dispositivo de ASP - O nome do dispositivo de ASP que contém o receptor de diário.
QEZJRCVLIB	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações efectuadas no diário ao nome da biblioteca que contém o receptor de diário. Este campo está em branco se o objecto nunca tiver sido registado em diário.
QEZJRCVNAM	GRAPHIC(10)	Foi necessário aplicar com êxito as alterações efectuadas ao diário no receptor de diário mais antigo. Quando o campo Aplicar informações está definido para PARTIAL_TRANSACTION o receptor de diário inicia a transacção parcial. De outro modo; o receptor de diário inicia a operação de gravação. Este campo está em branco se o objecto nunca tiver sido registado em diário.
QEZJRNID	GRAPHIC (10)	Este campo associa o objecto a ser registado em diário a um identificador que pode ser utilizado em vários comandos relacionados com o registo em diário e APIs. Este campo está em branco se o objecto nunca tiver sido registado em diário.
QEZJRNLIB	GRAPHIC (10)	Se o valor do estado do registo em diário for JOURNALED, este campo contém o nome da biblioteca com o diário utilizado actualmente. Se o valor do estado do registo em diário for NOT_JOURNALED, este campo contém o nome da biblioteca com o último diário utilizado. Este campo está em branco se o objecto nunca tiver sido registado em diário.
QEZJRNNAM	GRAPHIC (10)	Se o valor do estado do registo em diário for JOURNALED, este campo contém o nome do diário utilizado actualmente. Se o valor do estado do registo em diário for NOT_JOURNALED, este campo contém o nome do último diário utilizado para este objecto. Este campo está em branco se o objecto nunca tiver sido registado em diário.
QEZJRNSTR	TIMESTAMP	Os segundos decorridos desde o início da fase correspondente à última data e hora de registo em diário do objecto. Este campo apresenta o valor NULL se o objecto nunca tiver sido registado em diário.
QEZJRNSTS ¹	SMALLINT	O estado actual do registo em diário do objecto. Este campo terá um dos valores que se seguem: 0 (NOT_JOURNALED) - Actualmente não está a ser efectuado o registo em diário do objecto. 1 (JOURNALED) - Actualmente está a ser efectuado o registo em diário do objecto.
QEZJSUBTRE	SMALLINT	Se o sinalizador for devolvido, este objecto é um directório com uma semântica de subárvore de registo em diário do sistema de ficheiros integrado. 0 - O registo em diário do objecto não é efectuado com uma semântica de subárvore. 1 - O registo em diário do objecto é efectuado com uma semântica de subárvore. Os objectos novos criados nesta subárvore do directório herdam os atributos e opções do registo em diário deste directório.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZJTRNI	GRAPHIC (1)	<p>Este campo descreve as informações acerca do estado actual do objecto na medida em que este se encontra relacionado com limites de controlo de consolidações. Os valores válidos são:</p> <p>x'00' (NONE) - Não existem transacções parciais.</p> <p>x'01' (PARTIAL_TRANSACTION) - O objecto foi restaurado com as transacções parciais. Só é possível utilizar este objecto se utilizar o comando Aplicar alterações registadas em diário (APYJRNCHG) ou Remover alterações registadas em diário (RMVJRNCHG) para executar ou remover as alterações efectuadas às transacções parciais.</p> <p>x'02' (ROLLBACK_ENDED) - Foi concluída a operação de remoção de alterações do objecto com a opção "Terminar Remoção de Alterações" do ecrã Trabalhar com Definição de Consolidação (WRKCMTDFN). Recomenda-se que restaure o objecto uma vez que não o pode utilizar. Como última opção, é possível utilizar o comando Alterar objecto registado em diário (CHGJRNOBJ) para permitir a utilização do objecto. No entanto, desta forma, o objecto pode ficar num estado inconsistente.</p>
QEZLANGID	GRAPHIC (3)	Um ID com três caracteres que representa a linguagem em que se encontra o nome do objecto (campo QEZOBJNAM).
QEZLOCAL	SMALLINT	<p>Se um objecto é armazenado localmente ou num sistema remoto. Se um objecto é local ou remoto varia de acordo com as respectivas regras do sistema de ficheiros. Os objectos do sistema de ficheiros que não contenham um indicador remoto ou local são considerados remotos. Os valores válidos são:</p> <p>1 - Os dados do objecto estão armazenados localmente.</p> <p>2 - Os dados do objecto estão num sistema remoto.</p>
QEZMLTSIG	SMALLINT	<p>Se um objecto tem mais do que uma assinatura digital do i5/OS. Os valores válidos são:</p> <p>0 - O objecto tem apenas uma assinatura digital.</p> <p>1 - O objecto tem mais do que uma assinatura digital. Se o campo QEZSYSSIG tiver o valor 1, pelo menos uma das assinaturas é considerada fidedigna pelo sistema.</p>
QEZMODE	INTEGER	O tipo e o modo de acesso ao ficheiro. Para obter mais informações sobre o modo, consulte a API open() API.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZMSTGOPT	SMALLINT	<p>Esta opção deverá ser utilizada para determinar a atribuição e utilização de memória principal pelo sistema para o objecto especificado. Esta opção só poderá ser especificada para ficheiros de dados contínuos nos sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador. Os valores válidos são:</p> <p>0 - A memória principal será atribuída normalmente. Ou seja, será atribuído e utilizado o máximo de memória principal possível. Desta forma, é reduzido o número de operações de I/O do disco, uma vez que as informações são armazenadas na cache da memória principal.</p> <p>1 - A memória principal será atribuída para reduzir o espaço utilizado pelo objecto. Ou seja, será atribuído e utilizado o mínimo de memória principal possível. Desta forma, é reduzida a utilização da memória principal, ao mesmo tempo que aumenta o número de operações de I/O do disco, uma vez que são armazenadas menos informações na cache da memória principal.</p> <p>2 - O sistema determinará de forma dinâmica a atribuição de memória principal ideal para o objecto consoante as outras actividades do sistema e conflitos da memória principal. Ou seja, se existirem poucos conflitos na memória principal, será atribuído e utilizado o máximo de memória possível de forma a reduzir o número de operações de I/O do disco. E quando existem conflitos significativos da memória principal, será atribuída e utilizada menos memória principal para reduzi-los. Esta opção só tem efeito quando a opção de paginação do conjunto de memória é *CALC. Sempre que a opção de paginação do conjunto de memória é *FIXED, o comportamento é o mesmo que STG_NORMAL. Se aceder ao objecto através de um servidor de ficheiros, esta opção não tem qualquer efeito. Em vez disso, tem um comportamento idêntico a STG_NORMAL.</p>
QEZNLNK	INTEGER	O número de ligações permanentes ao objecto.
QEZNMCCSID	INTEGER	O CCSID no qual o nome do objecto (campo QEZOBJNAM) está representado.
QEZNONSAV	SMALLINT	<p>Se pode ou não guardar o objecto. Os valores válidos são:</p> <p>0 - O objecto será guardado.</p> <p>1 - O objecto não será guardado. Além disso, se este objecto for um directório, nenhum dos objectos da subárvore do directório será guardado a menos que estejam especificados explicitamente como um objecto a guardar. A subárvore inclui todos os subdirectórios e os respectivos objectos.</p>
QEZOBJLEN	INTEGER	O número de bytes existentes no nome do objecto (campo QEZOBJNAM).
QEZOBJNAM ¹	VARGRAPHIC (1024)	O nome do objecto. ²
QEZOBJTYPE ¹	GRAPHIC (10)	O tipo de objecto.
QEZOFLOW	SMALLINT	<p>Indica se o objecto excedeu o conjunto de memória auxiliar em que se encontra. Os valores válidos são:</p> <p>0 - O conjunto de memória auxiliar não foi excedido.</p> <p>1 - O conjunto de memória auxiliar foi excedido.</p>

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

Nome do campo	Tipo de campo	Descrição do campo
QEZOWN ¹	GRAPHIC (10)	O nome do perfil de utilizador que é o proprietário do objecto ou o seguinte valor especial: *NOUSRPRF - Este valor especial é utilizado pelo Network File System para indicar que não existe um perfil de utilizador no servidor iSeries local com um ID de utilizador (UID) que corresponda ao UID do objecto remoto.
QEZOWNPGP	GRAPHIC (10)	O nome do perfil de utilizador que é o grupo principal do objecto ou os seguintes valores especiais: *NONE - O objecto não tem um grupo principal. *NOUSRPRF - Este valor especial é utilizado pelo Network File System para indicar que não existe um perfil de utilizador no servidor local com um ID de grupo (GID) correspondente ao GID do objecto remoto.
QEZPCARC	SMALLINT	Se o objecto foi alterado desde a última verificação. 0 - O objecto não foi alterado. 1 - O objecto foi alterado.
QEZPCHID ¹	SMALLINT	Se o objecto pode ser apresentado utilizando uma listagem de directórios comum. 0 - O objecto não está oculto. 1 - O objecto está oculto.
QEZPCREAD	SMALLINT	Se o objecto pode ser escrito ou eliminado, se os respectivos atributos expandidos podem ser alterados ou eliminados ou o tamanho do objecto alterado. Os valores válidos são: 0 - O objecto pode ser alterado. 1 - O objecto não pode ser alterado.
QEZPCSYS	SMALLINT	Se o objecto é um ficheiro de sistema e é excluído nas procuras de directórios normais. 0 - O objecto não é um ficheiro de sistema. 1 - O objecto é um ficheiro de sistema.
QEZPRMLNK	SMALLINT	Quando um objecto tem vários nomes, este campo é definido apenas para o primeiro nome encontrado.
QEZRDEV	BIGINT	Se o objecto representa um ficheiro especial de um dispositivo, representa o dispositivo real.
QEZREGION	GRAPHIC (2)	Um ID com dois caracteres que representa o país do nome do objecto (campo QEZOBJNAM). Este ID afecta as acções cuja tendência é serem definidas pela localização da acção, tais como, a sequência de ordenação.
QEZSBINARY	GRAPHIC (1)	Indica se o objecto foi pesquisado no modo binário, quando foi pesquisado anteriormente. Este campo terá um dos valores que se seguem: x'00' - O objecto não foi pesquisado no modo binário. x'01' - O objecto foi pesquisado no modo binário. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito no modo binário. Se o estado de pesquisa do objecto é SCAN_FAILURE, ocorreu uma falha na pesquisa do objecto no modo binário.

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

Nome do campo	Tipo de campo	Descrição do campo
QEZSCCSID1	INTEGER	Indica se o objecto foi pesquisado no CCSID listado, quando foi pesquisado anteriormente. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito neste CCSID. Se o estado de pesquisa do objecto é SCAN_FAILURE, ocorreu uma falha na pesquisa do objecto neste CCSID. Um valor 0 significa que este campo não é aplicável.
QEZSCCSID2	INTEGER	Indica se o objecto foi pesquisado no CCSID listado, quando foi pesquisado anteriormente. Se o estado de pesquisa do objecto é SCAN_SUCCESS, a pesquisa do objecto foi efectuada com êxito neste CCSID. Se o estado de pesquisa do objecto é SCAN_FAILURE, o valor deste campo será 0. Um valor 0 significa que este campo não é aplicável.
QEZSCN	GRAPHIC (1)	<p>Se o objecto será pesquisado quando os programas de saída são registados através de um dos pontos de saída relacionados com a leitura do sistema de ficheiros integrado.</p> <p>Os valores válidos são:</p> <p>x'00' (SCANNING_NO) - O objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p> <p>x'01' (SCANNING_YES) - O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.</p> <p>x'02' (SCANNING_CHGONLY) - O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa apenas se o objecto tiver sido modificado desde a última pesquisa efectuada ao objecto. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado. Este atributo só entrará em vigor se o valor do sistema Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) tiver *USEOCOATR especificado. De outro modo, será considerado como se o atributo fosse SCANNING_YES. Nota: Se o valor Pesquisa de controlo dos sistemas de ficheiros (QSCANFCTL) *NOPOSTRST não for especificado quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.</p>
QEZSIG ¹	SMALLINT	<p>Se um objecto tem uma assinatura digital do i5/OS. Os valores válidos são:</p> <p>0 - O objecto não tem uma assinatura digital do i5/OS.</p> <p>1 - O objecto tem uma assinatura digital do i5/OS.</p>

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

Nome do campo	Tipo de campo	Descrição do campo
QEZSSIGDF	GRAPHIC (1)	<p>As assinaturas de pesquisa indicam o nível de suporte do software de pesquisa.</p> <p>Quando um objecto está num grupo de ASPs independentes, a assinatura de pesquisa do objecto é comparada à assinatura de pesquisa do grupo de ASPs independentes associado. Quando um objecto não está num grupo de ASPs independentes, a assinatura de pesquisa do objecto é comparada com o valor da assinatura de pesquisa global. Este campo terá um dos valores que se seguem:</p> <p>x'00' - As assinaturas comparadas não são diferentes.</p> <p>x'01' - As assinaturas comparadas são diferentes.</p>
QEZSTATUS	GRAPHIC (1)	<p>O estado de pesquisa associado a este objecto. Este campo terá um dos valores que se seguem:</p> <p>x'00' (SCAN_REQUIRED) - É necessário efectuar uma pesquisa do objecto porque ainda não foi pesquisado pelos programas de saída relacionados com a pesquisa ou porque os dados dos objectos ou CCSID foram modificados desde a última pesquisa. Seguem-se exemplos de modificações de dados de objectos ou CCSID: escrever num objecto, directamente ou através da definição da memória; truncar o objecto; limpar o objecto; e alterar o atributo CCSID dos objectos.</p> <p>x'01' (SCAN_SUCCESS) - O objecto foi pesquisado através de um programa de saída relacionado com a pesquisa e no último pedido de pesquisa esta foi efectuada com êxito.</p> <p>x'02' (SCAN_FAILURE) - O objecto foi pesquisado através de um programa de saída relacionado com a pesquisa e no último pedido de pesquisa não foi possível efectuar a pesquisa e a operação não foi concluída. Se ocorreu uma falha num objecto, este só será pesquisado novamente se a assinatura de pesquisa do objecto for diferente da assinatura chave de pesquisa global ou da assinatura chave de pesquisa de grupo de ASPs independentes, conforme for adequada. Assim, não será possível executar os pedidos seguintes para trabalhar com o objecto e será apresentada uma indicação de falha na pesquisa. Exemplos de pedidos com falha são: abrir o objecto, alterar o CCSID do objecto, copiar o objecto.</p> <p>x'05' (SCAN_PENDING_CVN) - O objecto não está num directório *TYPE2 e, por conseguinte, só será pesquisado após a conversão do directório.</p> <p>x'06' (SCAN_NOT_REQUIRED) - Não é necessário efectuar uma pesquisa do objecto porque este está assinalado para não ser pesquisado.</p>
QEZSTGFREE ¹	SMALLINT	<p>Se os dados do objecto foram movidos para offline, libertando memória online. Os valores válidos são:</p> <p>0 - Os dados do objecto não estão offline.</p> <p>1 - Os dados do objecto estão offline.</p>
QEZSYSARC	SMALLINT	<p>Se o objecto foi alterado e precisa de ser guardado. Está activado para quando a hora de alteração do objecto é actualizada e desactivado caso o objecto já tenha sido guardado.</p> <p>0 - O objecto não foi alterado e não necessita de ser guardado.</p> <p>1 - O objecto foi alterado e necessita de ser guardado.</p>

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

Nome do campo	Tipo de campo	Descrição do campo
QEZYSSIG	SMALLINT	Se o objecto foi assinado por uma fonte considerada fidedigna pelo sistema. Os valores válidos são: 0 - Nenhuma das assinaturas é de uma fonte considerada fidedigna pelo sistema. 1 - O objecto foi assinado por uma fonte considerada fidedigna pelo sistema. Se o objecto tem várias assinaturas, pelo menos uma das assinaturas é de uma fonte considerada fidedigna pelo sistema.
QEZUDATE	TIMESTAMP	Os segundos decorridos desde a fase correspondente à data da última utilização do objecto. O valor deste campo é zero quando o objecto é criado. Se os dados de utilização não forem mantidos para o tipo i5/OS nem para o sistema de ficheiros ao qual pertence um objecto, o valor deste campo é zero.
QEZUDCOUNT	INTEGER	O número de dias de utilização de um objecto. Existem significados diferentes para a utilização de acordo com o sistema de ficheiros específico e com os tipos de objecto individuais suportados num sistema de ficheiros. A utilização pode indicar a abertura ou fecho de um ficheiro ou pode referir-se à adição de ligações, mudança de nome, restauro ou a dar saída a um objecto. Esta contagem aumenta a cada dia que um objecto seja utilizado e o valor zero é repostado se chamar a API Qp0lSetAttr().
QEZUDFTYP2	SMALLINT	O formato de ficheiro predefinido dos ficheiros de dados contínuos (*STMF) criados no sistema de ficheiros definido pelo utilizador. Os valores válidos são: 0 - O formato de ficheiro de dados contínuos é *TYPE1. 1 - O formato de ficheiro de dados contínuos é *TYPE2.
QEZUID	INTEGER	Todos os utilizadores do sistema têm de ter um número de identificação de utilizador numérico exclusivo (UID).
QEZURESET	INTEGER	Os segundos decorridos desde a última vez que a fase correspondente à contagem dos dias da data foi colocada a zero (0). Esta data é definida com a data actual quando é chamada a API Qp0lSetAttr() para repor a contagem dos dias de utilização como zero.
Notas:		
<ol style="list-style-type: none"> 1. Este campo está incluído no subconjunto de campos utilizado pelo comando PRDIRINF. 2. Neste campo, só o nome do objecto é memorizado. O resto do nome do caminho é armazenado no campo QEZDIRNAM1 se o comprimento do nome do directório estiver abaixo de 1 KB (equivalente a 1024 bytes) ou em QEZDIRNAM2 se os nomes de directório estiverem acima de 1 KB (equivalente a 1024 bytes). 		

A tabela que se segue é um exemplo de uma tabela que lista os directórios processados pelo comando RTVDIRINF.

Tabela 9. QAEZDxxxD (armazenar atributos de directórios)

Nome do campo	Tipo de campo	Descrição do campo
QEZDFID	INTEGER	O ID de ficheiro do directório.
QEZDIRFID	GRAPHIC (16)	O ID de ficheiro do directório. Um identificador associado ao objecto. É possível utilizar um ID de ficheiro com o Qp0lGetPathFromFileID() para obter um nome de caminho do objecto.
QEZDIRFSID	BIGINT	O ID do sistema de ficheiros do directório.
QEZDIRGID	BIGINT	O ID de geração.

Tabela 9. QAEZDxxxD (armazenar atributos de directórios) (continuação)

Nome do campo	Tipo de campo	Descrição do campo
QEZDIRIDX	INTEGER	Identificador do nome do caminho (só aplicável aos directórios).
QEZDIRLEN ¹	INTEGER	O comprimento do nome do caminho do directório.
QEZDIRNAM1 ¹	VARGRAPHIC (1024)	O caminho do directório ascendente. Utilizado apenas quando o comprimento do caminho está abaixo de 1 KB (1024 bytes).
QEZDIRNAM2 ¹	DBCLOB (16M)	O caminho do directório ascendente. Utilizado apenas quando o comprimento do caminho está acima de 1 KB (1024 bytes). É possível memorizar caminhos até 16 MB de comprimento.
QEZDRCCSID	INTEGER	O CCSID do directório.
QEZDREGION	GRAPHIC (2)	O ID da região do caminho do directório.
QEZLANGID	GRAPHIC (3)	O ID do idioma do caminho do directório.
Nota:		
<ul style="list-style-type: none"> • Este campo está incluído no subconjunto de campos utilizado pelo comando PRTDIRINF. 		

A tabela que se segue mostra as informações armazenadas pelo comando RTVDIRINF no que respeita aos ficheiros que criou ao ser executado. Se o ficheiro que contém estas informações não existir, o comando RTVDIRINF cria o mesmo; quando o comando for executado posteriormente, as informações são anexadas ao ficheiro existente. O comando PRTDIRINF utiliza estas informações para determinar quais os ficheiros de base de dados utilizados para armazenar as informações obtidas pelas diferentes instâncias do comando RTVDIRINF.

Tabela 10. QUSRSYS/QAEZDBFILE (armazenar ficheiros criados)

Nome do campo	Tipo de campo	Descrição do campo
QEZDIRFILE ¹	VARGRAPHIC (20)	O nome do ficheiro gerado para armazenar os índices remissivos do directório.
QEZDIRSRC	VARGRAPHIC (5000)	Caminho especificado no parâmetro DIR (RTVDIRINF).
QEZENDTIME	TIMESTAMP	Data/Hora de conclusão do RTVDIRINF.
QEZLIB ¹	VARGRAPHIC (20)	A biblioteca em que se encontram ambos os ficheiros gerados.
QEZOBJFILE ¹	VARGRAPHIC (20)	O nome do ficheiro gerado para armazenar os atributos do objecto.
QEZPLANGID	GRAPHIC (3)	O ID do idioma do caminho.
QEZPRCCSID	INTEGER	O caminho de CCSID.
QEZPREGION	GRAPHIC (2)	O ID da região do caminho.
QEZSTRTIME	TIMESTAMP	Data/Hora de submissão do RTVDIRINF.
Nota:		
<ul style="list-style-type: none"> • Este campo está incluído no subconjunto de campos utilizado pelo comando PRTDIRINF. 		

Informações relacionadas

Comando Obter informações do directório (RTVDIRINF)

API Qp01GetPathFromID()

API Qp01SetAttr()

Comando Aplicar alterações registadas em diário (APYJRNCHG)

Comando Remover alterações registadas em diário (RMVJRNCHG)

Comando Alterar objecto registado em diário (CHGJRNOBJ)

Comando Imprimir informações do directório (PRTDIRINF)

Aceder a dados de RTVDIRINF:

Existem várias opções para acesso aos dados das tabelas.

Abaixo encontra-se uma lista com formas de acesso a dados criados utilizando o comando Obter informações de directório (RTVDIRINF):

- Utilizar o comando Imprimir informações de directório (PRTDIRINF)

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

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

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

Por exemplo, se pretender seleccionar objectos num caminho específico (anteriormente recolhidos utilizando o comando RTVDIRINF) com um tamanho de alocação superior a 10 KB, pode executar uma consulta como a que se segue:

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

- É possível criar os seus próprios programas e aceder às tabelas da base de dados através de quaisquer métodos DB válidos.

Informações relacionadas

Comando Imprimir informações do directório (PRTDIRINF)

Comando Iniciar SQL (STRSQL)

Instruções SQL incorporadas

CLIS de SQL

Utilizar dados de RTVDIRINF:

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

- No caso da Tabela 8 na página 80, pode efectuar consultas para criar relatórios ou estatísticas com base em qualquer um dos campos desta tabela. O comando PRTDIRINF não inclui relatórios com base em todos os campos. Em vez disso, é utilizado um subconjunto.
- Os dados da Tabela 9 na página 90 contêm todos os directórios existentes no caminho especificado no parâmetro DIR do comando RTVDIRINF. Se pretende saber atributos específicos acerca do nome de caminho, por exemplo o CCSID, o ID do idioma ou comprimento, estes dados serão úteis. Todos os directórios armazenados nesta tabela têm também um valor exclusivo ou índice remissivo que os identifica. Na Tabela 8 na página 80, pode encontrar o mesmo campo, QEZDIRIDX, que indicará os objectos que pertencem a cada directório. Para saber quais os objectos que pertencem a um directório, pode efectuar uma consulta utilizando junções. Por exemplo, a instrução de consulta que se segue selecciona os nomes de todos os objectos existentes no directório "/MEUDIR":

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

- A Tabela 10 na página 91 é utilizada na maior parte das vezes pelo comando PRTDIRINF para obter dados específicos sobre execuções de RTVDIRINF. Exemplos desta situação são: os nomes das tabelas criadas, a biblioteca em que se encontram as tabelas e a hora de início e de fim do processamento. Poderá utilizar esta tabela para saber quando foi emitido um comando RTVDIRINF ou em que tabelas procurar para efectuar uma consulta.

Aceder utilizando APIs

Pode utilizar interfaces de programas de aplicações (API - Application Program Interfaces) para aceder ao sistema de ficheiros integrado.

Referências relacionadas

“Executar operações utilizando APIs” na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Aceder utilizando o iSeries Navigator

O iSeries Navigator é a interface gráfica do utilizador que permite gerir e administrar os sistemas a partir da área de trabalho do Windows. O iSeries Navigator torna as operações e administração do sistema mais fáceis e mais produtivas.

Por exemplo, pode copiar um perfil de utilizador para outro sistema arrastando-o de um servidor do iSeries para outro servidor do iSeries. Os assistentes dão indicações para a configuração da segurança e de serviços e aplicações de TCP/IP.

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

Trabalhar com ficheiros e pastas

- “Criar uma pasta” na página 136
- “Remover uma pasta” na página 136
- “Dar entrada de um ficheiro” na página 136
- “Dar saída de um ficheiro” na página 136
- “Definir permissões” na página 138
- “Configurar conversão de texto do ficheiro” na página 138
- “Enviar um ficheiro ou uma pasta para outro sistema” na página 139
- “Alterar opções para a definição do pacote” na página 139
- “Marcar uma data e uma hora para enviar o ficheiro ou a pasta” na página 140
- “Definir se os objectos devem ou não ser pesquisados” na página 142

Trabalhar com partilhas de ficheiros

- “Criar uma partilha de ficheiros” na página 140
- “Alterar uma partilha de ficheiros” na página 140

Trabalhar com sistemas de ficheiros definidos pelo utilizador

- “Criar um novo sistema de ficheiros definido pelo utilizador” na página 141
- “Montar sistema de ficheiros definido pelo utilizador” na página 141
- “Desmontar sistema de ficheiros definido pelo utilizador” na página 141

Registar objectos em diário

- “Iniciar registo em diário” na página 107
- “Terminar registo em diário” na página 108

Referências relacionadas

“Aceder utilizando um PC” na página 95

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

Aceder utilizando o iSeries NetServer

iSeries Support for Windows Network Neighborhood (iSeries NetServer) é uma função do i5/OS que permite aos clientes do Windows aceder a caminhos de directório partilhados e a filas de saída partilhadas do i5/OS. O iSeries NetServer permite que os PCs com o software do Windows acedam de forma contínua aos dados e impressoras geridos pelo iSeries.

Os clientes de PC numa rede utilizam as funções de partilha de ficheiros e impressão incluídas nos respectivos sistemas operativos. Ou seja, não necessita de instalar software adicional no PC para poder utilizar o iSeries NetServer.

Os clientes do Linux com o software de cliente Samba instalado também podem aceder aos dados e impressoras de forma contínua através do iSeries NetServer. Os directórios do iSeries NetServer partilhados podem ser montados em clientes Linux como sistemas de ficheiros Samba (smbfs) de uma forma semelhante à montagem de sistemas de ficheiros NFS que tenham sido exportados a partir do iSeries.

Uma partilha de ficheiros do iSeries NetServer é um caminho de directório que o iSeries NetServer partilha com clientes na rede do iSeries. Uma partilha de ficheiros pode ser constituída por qualquer directório do sistema de ficheiros integrado do iSeries. Para poder trabalhar com a partilha de ficheiros utilizando o iSeries NetServer, tem de criar uma partilha de ficheiros do iSeries NetServer e, se necessário, alterar uma partilha de ficheiros do iSeries NetServer utilizando o iSeries Navigator.

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

1. Faça clique com o botão direito do rato sobre **Iniciar** e seleccione **Explorador** para abrir o Explorador do Windows no seu PC com o Windows.
2. Abra o menu Ferramentas (Tools) e seleccione **Mapear unidade de rede (Map network drive)**.
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 em **OK**.

Nota: Ao estabelecer ligação através do iSeries NetServer, o nome do servidor pode ser diferente do nome utilizado pelo Família 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 Família 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/rui com o nome dirrui, um utilizador poderá aceder a esse directório a partir de um PC com o nome \\QAS400X\DIRRUI, ou a partir de um cliente do LINUX com o nome //qas400x/dirrui.

O sistema de ficheiros "raiz" (/) fornece um rendimento muito mais elevado para a assistência a ficheiros de PC do que outros sistemas de ficheiros do iSeries. Pode desejar mover ficheiros para o sistema de ficheiros "raiz" (/). Consulte "Mover ficheiros ou pastas para outro sistema de ficheiros" na página 137 para obter mais informações.

Referências relacionadas

"Aceder utilizando um PC" na página 95

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

Informações relacionadas

iSeries NetServer

Partilhas de ficheiros do iSeries NetServer

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

Acesso através do Protocolo de Transferência de Ficheiros

O cliente do Protocolo de Transferência de Ficheiros (FTP - File Transfer Protocol) permite transferir ficheiros localizados no servidor iSeries, incluindo nos sistemas de ficheiros "raiz" (/), QOpenSys, QSYS.LIB, QSYS.LIB de ASP independente, QOPT e QFileSvr.400.

Também permite transferir pastas e documentos do sistema de ficheiros dos serviços da 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)
- Network File System (NFS)
- Sistema de ficheiros NetWare (QNetWare)
- Sistema de ficheiros do iSeries NetClient (QNTC)

No entanto, tenha em consideração as seguintes restrições:

- O sistema de ficheiros integrado limita o suporte de FTP exclusivamente à transferência de dados de ficheiros. Não pode utilizar o FTP para transferir dados de atributos.
- Os sistemas de ficheiros QSYS.LIB e QSYS.LIB de ASP independente limitam o suporte de FTP a membros de ficheiros físicos, membros de ficheiros 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.

Referências relacionadas

"Aceder utilizando um PC"

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

Informações relacionadas

FTP

Transferir ficheiros utilizando FTP

Aceder utilizando um PC

Se o seu PC estiver ligado a um servidor 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 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 Família iSeries Access pode ser configurado para executar automaticamente esta conversão e pode ainda especificar que a conversão seja executada nos ficheiros que tenham uma extensão específica.

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

Se estiver ligado a um servidor iSeries através do PC, o sistema de ficheiros integrado disponibiliza os directórios e objectos do servidor 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, um cliente de FTP ou o iSeries Navigator (um componente do Família iSeries Access). O PC utiliza clientes de partilha de ficheiros do Windows para aceder ao iSeries NetServer, que é executado no servidor iSeries.

Conceitos relacionados

“Aceder utilizando o iSeries Navigator” na página 93

O iSeries Navigator é a interface gráfica do utilizador que permite gerir e administrar os sistemas a partir da área de trabalho do Windows. O iSeries Navigator torna as operações e administração do sistema mais fáceis e mais produtivas.

Tarefas relacionadas

“Aceder utilizando o iSeries NetServer” na página 94

iSeries Support for Windows Network Neighborhood (iSeries NetServer) é uma função do i5/OS que permite aos clientes do Windows aceder a caminhos de directório partilhados e a filas de saída partilhadas do i5/OS. O iSeries NetServer permite que os PCs com o software do Windows acedam de forma contínua aos dados e impressoras geridos pelo iSeries.

Referências relacionadas

“Acesso através do Protocolo de Transferência de Ficheiros” na página 95

O cliente do Protocolo de Transferência de Ficheiros (FTP - File Transfer Protocol) permite transferir ficheiros localizados no servidor iSeries, incluindo nos sistemas de ficheiros "raiz" (/), QOpenSys, QSYS.LIB, QSYS.LIB de ASP independente, QOPT e QFileSvr.400.

Converter directórios *TYPE1 em *TYPE2

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original. Os directórios *TYPE2 têm uma estrutura interna diferente dos directórios *TYPE1 e facultam um aumento do rendimento e fiabilidade.

Pouco depois do i5/OS V5R3M0 ou de uma edição posterior ser instalada, a conversão para directórios *TYPE2 é iniciada automaticamente em qualquer dos sistemas de ficheiros que ainda não tenham sido convertidos, tendo em vista o suporte de directórios *TYPE2. Esta conversão não deverá afectar as actividades do sistema.

Conceitos relacionados

“Directórios *TYPE2” na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

Descrição geral da conversão de *TYPE1 em *TYPE2

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original. Os directórios *TYPE2 têm uma estrutura interna diferente dos directórios *TYPE1 e facultam um aumento do rendimento e fiabilidade. Além do aumento de rendimento e fiabilidade, as novas funções, tais como o suporte de pesquisa do sistema de ficheiros integrado, só estão disponíveis para objectos existentes nos directórios *TYPE2. Consulte "Suporte de pesquisa" na página 21 para obter mais informações.

Pouco depois do sistema operativo i5/OS V5R3M0 ou de uma edição posterior ser instalada, a conversão para directórios *TYPE2 é iniciada automaticamente em qualquer dos sistemas de ficheiros que ainda não tenham sido convertidos, tendo em vista o suporte de directórios *TYPE2. Esta conversão não deverá afectar as actividades do sistema uma vez que será executada como um trabalho em segundo plano de prioridade baixa.

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

Os sistemas de ficheiros elegíveis para este tipo de conversão automática são os sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador para ASPs de 1 a 32.

Nota: Pode impedir a conversão automática para directórios *TYPE2 se converter os sistemas de ficheiros antes de instalar o sistema operativo V5R3M0 ou uma edição posterior.

Conceitos relacionados

"Directórios *TYPE2" na página 10

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File System), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2. O formato de directório *TYPE2 é um melhoramento do formato de directório *TYPE1 original.

Referências relacionadas

"Determinar o estado de conversão"

Pouco depois do sistema operativo i5/OS V5R3M0 ou de uma edição posterior ser instalada, a conversão para directórios *TYPE2 é iniciada automaticamente em qualquer dos sistemas de ficheiros que ainda não tenham sido convertidos, tendo em vista o suporte de directórios *TYPE2. Este processamento de conversão ocorrerá num módulo secundário de trabalho do sistema QFILESYS1.

"Sugestões: ASP independente" na página 101

Se os sistemas de ficheiros definidos pelo utilizador numa ASP independente ainda não tiverem sido convertidos para o formato de directório *TYPE2, serão convertidos na primeira vez que a ASP independente for activada num sistema com a versão V5R2 ou posterior do sistema operativo instalada.

Considerações sobre conversão

Seguem-se vários aspectos a ter em consideração durante o processo de conversão.

Determinar o estado de conversão

Pouco depois do sistema operativo i5/OS V5R3M0 ou de uma edição posterior ser instalada, a conversão para directórios *TYPE2 é iniciada automaticamente em qualquer dos sistemas de ficheiros que ainda não tenham sido convertidos, tendo em vista o suporte de directórios *TYPE2. Este processamento de conversão ocorrerá num módulo secundário de trabalho do sistema QFILESYS1.

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

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

Uma vez que o trabalho QFILESYS1 está a processar a conversão, pode apresentar o registo de trabalhos QFILESYS1 de mensagens indicando a existência de problemas na conversão. Além disso, são enviadas várias mensagens de progressão sobre as conversões do sistema de ficheiros. Estas mensagens incluem informações como: o sistema de ficheiros a ser convertido, o número de ligações processadas no sistema de ficheiros, a percentagem de directórios processados no sistema de ficheiros, etc. Todas as mensagens de erro e muitas das mensagens de progresso são também enviadas para a fila de mensagens QSYSOPR. Deste modo, para uma consulta futura, é recomendado certificar-se de que mantém os registos QHST ou os registos de trabalhos QFILESYS1, que contém estas mensagens. Quando os sistemas de ficheiros tiverem sido totalmente convertidos e o sistema de ficheiros integrado estiver a funcionar conforme previsto, pode eliminar estas informações do histórico.

Conceitos relacionados

"Descrição geral da conversão de *TYPE1 em *TYPE2" na página 97

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

Informações relacionadas

Comando Converter directório (CVTDIR)

Criar perfis de utilizador

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

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

Informações relacionadas

"Alterar o proprietário de um directório" na página 99

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.

Objectos com o nome mudado

Os directórios *TYPE2 requerem que os nomes de ligações sejam nomes UTF-16 válidos.

A regra de nomenclatura de directórios *TYPE2 difere dos directórios *TYPE1, que têm nomes UCS2 de Nível 1. Por este motivo, podem ser encontrados nomes inválidos ou duplicados durante uma conversão de directórios. Quando é encontrado um nome inválido ou duplicado, ele é alterado para um nome UTF-16 exclusivo, válido e é enviada a mensagem CPIA08A para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR, apresentando o nome original e o novo nome. Caracteres combinados ou pares de caracteres delegados inválidos contidos num nome podem implicar a mudança de nome de um objecto.

Para obter mais informações sobre UTF-16, consulte a página inicial da Unicode (www.unicode.org ).

Caracteres combinados:

Alguns caracteres podem ser constituídos por mais do que um carácter Unicode.

Por exemplo, os caracteres com um acento (por exemplo, é ou à) ou com um trema (por exemplo, ä ou ö) têm de ser alterados ou *normalizados*, num formato comum para que possam ser armazenados no directório, de modo a que todos os objectos tenham um nome único. 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.

Delegar caracteres:

Alguns caracteres não têm representação válida em Unicode.

Estes caracteres têm alguns valores especiais; são formados por dois caracteres Unicode em dois intervalos específicos de modo a que o primeiro carácter Unicode esteja num intervalo (por exemplo, 0xD800-0xD8FF) e o segundo carácter Unicode esteja no segundo intervalo (por exemplo, 0xDC00-0xDCFF). A este par chama-se par substituto.

Se um dos caracteres Unicode faltar ou estiver fora da sequência, (apenas um carácter parcial), será um nome inválido. Foram permitidos nomes deste tipo nos directórios *TYPE1, mas não são permitidos em directórios *TYPE2. Para que a função de conversão continue, o nome é alterado antes de o objecto ser ligado ao directório *TYPE2, caso seja encontrado um nome que contenha um destes nomes inválidos.

Considerações sobre perfis de utilizador

Durante a execução da conversão, são efectuadas várias tentativas para assegurar que o mesmo perfil de utilizador que é o proprietário de directórios *TYPE1 continua a ser o proprietário dos directórios *TYPE2 correspondentes.

Uma vez que os directórios *TYPE1 e *TYPE2 existem momentaneamente ao mesmo tempo, essa situação afecta a quantidade de memória pertencente ao perfil de utilizador e o número de entradas no perfil de utilizador.

Alterar o máximo de memória para um perfil de utilizador:

Durante o processamento de conversão de directórios, existe momentaneamente e em simultâneo um determinado número de directórios em ambos os formatos que pertence ao mesmo perfil de utilizador.

Se durante o processamento da conversão for atingido o limite máximo de memória para o perfil de utilizador, o limite do perfil de utilizador será aumentado. A mensagem CPIA08C é enviada para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR.

Alterar o proprietário de um directório:

Se o perfil de utilizador ao qual pertence o directório *TYPE1 não puder ser o proprietário do directório *TYPE2 que é criado, o proprietário do directório *TYPE2 será definido como um dos perfis de utilizador alternativos.

A mensagem CPIA08B é enviada para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR e a conversão continua.

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. A mensagem CPIA08B é enviada para o registo de trabalhos QFILESYS1 e para a fila de mensagens QSYSOPR e a conversão continua.

Referências relacionadas

“Criar perfis de utilizador” na página 98

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

Requisitos de memória auxiliar

Devem ser considerados requisitos de memória auxiliar enquanto o sistema converte os directórios num sistema de ficheiros para o formato *TYPE2.

Seguem-se várias considerações relacionadas com requisitos de memória auxiliar:

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

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

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

Nota: Antes de instalar o sistema operativo i5/OS V5R3M0 ou uma edição posterior, poderá pretender executar a opção *ESTIMATE do comando (CVTDIR) do OS/400 V5R2, uma vez que pode fornecer uma estimativa moderada da quantidade de memória auxiliar necessária durante a conversão.

Informações relacionadas

Comando Converter directório (CVTDIR)

Sugestões: Ligação simbólica

Ligações simbólicas são objectos existentes no sistema de ficheiros integrado que contêm um caminho para outro objecto.

Existem algumas ocorrências durante a conversão quando o nome de um objecto pode ser alterado. Se um dos elementos do caminho numa ligação simbólica sofrer uma mudança de nome durante a conversão, o conteúdo da ligação simbólica deixará de apontar para o objecto.

Conceitos relacionados

“Ligação” na página 12

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.

Referências relacionadas

“Objectos com o nome mudado” na página 98

Os directórios *TYPE2 requerem que os nomes de ligações sejam nomes UTF-16 válidos.

Informações relacionadas

Estabelecer ligação simbólica (symlink())

Sugestões: ASP independente

Se os sistemas de ficheiros definidos pelo utilizador numa ASP independente ainda não tiverem sido convertidos para o formato de directório *TYPE2, serão convertidos na primeira vez que a ASP independente for activada num sistema com a versão V5R2 ou posterior do sistema operativo instalada.

Para fins de planeamento, é fornecida uma função de estimativa no OS/400 V5R1 para dar informações sobre o período de tempo que demorará uma conversão. Antes de activar a ASP independente no sistema operativo com a versão V5R2 ou posterior, execute a seguinte API no sistema operativo com a versão V5R1 quando a ASP independente (designado por ASP_NAME) for activada e estiver activa:

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

Conceitos relacionados

“Descrição geral da conversão de *TYPE1 em *TYPE2” na página 97

Os sistemas de ficheiros definidos pelo utilizador (UDFS - User-Defined File Systems), "raiz" (/) e QOpenSys no sistema de ficheiros integrado suportam o formato de directório *TYPE2 a partir do OS/400 V5R1.

Sugestões: Guardar e restaurar

Os directórios existentes como *TYPE1 podem ser guardados e restaurados num sistema de ficheiros que foi convertido em *TYPE2.

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

Sugestões: Regenerar objectos do sistema de ficheiros integrado

Enquanto o sistema estiver a converter os sistemas de ficheiros "raiz" (/), QOpenSys e UDFS de ASP do utilizador para suportarem o formato de directório *TYPE2, os comandos Regenerar armazenamento (RCLSTG) e Regenerar ligações de objectos (RCLLNK) não podem ser executados nos directórios do sistema de ficheiros integrado, incluindo os directórios pertencentes às ASPs independentes.

O valor do parâmetro OMIT(*DIR) pode ser utilizado no comando RCLSTG para omitir os directórios do sistema de ficheiros integrado e permitir que os objectos que não estejam relacionados com o sistema de ficheiros integrado sejam regenerados.

Conceitos relacionados

“Regenerar sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador” na página 108
A regeneração de sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador pode ser efectuada utilizando os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG).

Informações relacionadas

Comando Regenerar armazenamento (RCLSTG)

Comando Regenerar ligações de objectos (RCLLNK)

Pesquisa no sistema de ficheiros integrado

Os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e sistemas de ficheiros de utilizador UDFS ASP só serão pesquisados utilizando os pontos de saída relacionados com a pesquisa, quando os sistemas de ficheiros tiverem sido totalmente convertidos para o formato de directório *TYPE2.

Os atributos relacionados com a pesquisa podem ser definidos para objectos em directórios *TYPE1 e *TYPE2, de modo a especificar se os objectos devem ou não ser pesquisados, mesmo que o sistema de ficheiros não esteja totalmente convertido.

Enquanto o sistema estiver a converter objectos com o formato de directório *TYPE1 no formato de directório *TYPE2, o valor de sistema controlo de pesquisa Pesquisar no próximo acesso após restauro

do objecto (Scan on next access after object has been restored) é tido em consideração, como se o objecto convertido estivesse a ser restaurado. Por exemplo, se o valor Pesquisar no próximo acesso após restauro do objecto (Scan on next access after object has been restored) for especificado durante a conversão, um objecto existente num directório *TYPE1 e com o atributo o objecto não será pesquisado especificado, será pesquisado pelo menos uma vez, após a conversão total do sistema de ficheiros.

Conceitos relacionados

“Suporte de pesquisa” na página 21

Com o iSeries, pode pesquisar objectos do sistema de ficheiros integrado.

“Valores do sistema relacionados” na página 23

Existem dois valores de sistema relacionados com este suporte de pesquisa. Poderá utilizar estes dois valores de sistema para definir o ambiente de pesquisa que pretende para o servidor.

Registar objectos em diário

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

Estas informações fornecem uma breve descrição geral da gestão de diário, assim como considerações para registo de alterações de objectos do sistema de ficheiros integrado e uma descrição de suporte de registo de alterações para objectos do sistema de ficheiros integrado.

Informações relacionadas

Gestão de registos em diário

Descrição geral do registo em diário

Estes tópicos apresentam o suporte de registo de alterações para objectos do sistema de ficheiros integrado.

Informações relacionadas

Gestão de registos em diário

Gestão de registos em diário

O principal objectivo da gestão de registos em diário consiste em permitir que o utilizador recupere as alterações ocorridas num objecto, desde a última vez que este foi guardado.

Pode também utilizar a gestão de registos em diário para:

- Um registo de auditoria da actividade que ocorre em objectos do sistema
- Registar a actividade que ocorreu nos objectos diferentes dos que podem ser registados em diário
- Uma recuperação mais rápida ao restaurar a partir do suporte de guardar enquanto activo
- Auxiliar na replicação das alterações efectuadas ao objecto para outro sistema para obter uma maior disponibilidade ou equilíbrio da carga de trabalho
- Assistência nos testes aos programas de aplicação

Pode utilizar um registo em diário para definir os objectos que pretende proteger através da gestão de registos em diário. No sistema de ficheiros integrado pode registar ficheiros de dados contínuos, directórios e ligações simbólicas em diário. Apenas são suportados os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e UDFS.

Conceitos relacionados

“Objectos que deve registar em diário” na página 103

Existem certas questões a ter em consideração ao decidir se um determinado objecto do sistema de ficheiros integrado deve ser registado em diário.

Objectos que deve registar em diário

Existem certas questões a ter em consideração ao decidir se um determinado objecto do sistema de ficheiros integrado deve ser registado em diário.

Tem de ter em consideração as seguintes questões para determinar quais os objectos que é necessário registar em diário:

- 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.

Conceitos relacionados

“Gestão de registos em diário” na página 102

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.

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 de alterações do i5/OS.

Os tipos de objecto suportados são ficheiros de dados contínuos, os directórios e as 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 de alterações e terminar registo de alterações utilizando o iSeries Navigator, bem como apresentar informações sobre o estado do registo de alterações.

- l **Nota:** Os ficheiros de dados contínuos mapeados na memória, os ficheiros de volume virtual e os ficheiros de dados contínuos utilizados como espaços de armazenamento de rede do Servidor xSeries Integrado para iSeries (IXS) não podem ser registados em diário. Os directórios que podem conter objectos de ficheiro especial de blocos não podem ser registados em diário. Seguem-se alguns exemplos: /dev/QASP01, /dev/QASP22 e /dev/IASPNOME.

A lista que se segue resume o suporte de registo de alterações 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 de Alterações, Terminar de Alterações, Alterar Registo de Alterações e Aplicar Alterações Registadas em Diário, podem ser executados em subárvores completas de objectos do sistema de ficheiros integrado. Como opção, pode utilizar as listas de inclusão e exclusão que podem utilizar padrões de caracteres globais para nomes de objecto. Por exemplo, pode utilizar o comando Iniciar Registo em Diário para especificar o início em todos os objectos da árvore “/MinhaEmpresa” que correspondem ao padrão “*.data”, mas excluindo quaisquer objectos que correspondam aos padrões “A*.data” e “B*.data”.

- O suporte de registo em diário nos directórios inclui operações de directório, tais como adicionar ligações, remover ligações, criar, mudar o nome de e mover objectos dentro do directório.

Os directórios registados em diário suportam um atributo que pode ser definido para fazer com que novos objectos na subárvore herdem o actual estado de registo do directório. Quando este atributo é activado para um directório registado, o sistema inicia automaticamente o registo em diário de todos os ficheiros de dados contínuos, directórios e ligações simbólicas criados no ou ligados ao directório (através da adição de uma ligação permanente ou através da mudança do nome ou da movimentação do objecto).

Nota: Considerações de atributo de registo em diário herdado:

- Se mudar o nome de um objecto no mesmo directório em que reside actualmente, o registo em diário não será iniciado para o objecto, mesmo que o directório tenha o atributo de estado de registo em diário actual herdado activo.
 - Quando um directório é movido para um directório que tenha o atributo de registo em diário herdado activo, o registo em diário só será iniciado para o directório que foi movido se for apropriado. Os objectos dentro desse directório que foi movido não são afectados.
 - Se um objecto é restaurado para um directório que tenha o atributo de registo em diário herdado activo, o registo em diário do objecto não será iniciado se o mesmo nunca foi registado em diário.
 - Ao utilizar o comando Aplicar alterações registadas em diário (APYJRNCHG), o valor actual do atributo de registo de alterações herdado para os directórios não é utilizado. Em vez disso, é iniciado o registo em diário dos objectos criados como parte das aplicações ou que não são baseados no que aconteceu durante a actividade em tempo de execução que está a ser aplicada.
- Os nomes de objecto e os nomes completos de caminho estão contidos em várias entradas de diário de objectos do sistema de ficheiros integrado. Os nomes de objectos e de caminho estão activados para NLS (National Language Support).
 - Se o sistema abortar, é fornecida uma recuperação por carregamento do programa inicial (IPL - Initial Program Load) do sistema para os objectos do sistema de ficheiros integrado registados em diário.
 - O limite máximo de gravação suportado pelas várias interfaces de gravação é de 2 GB - 1. O tamanho máximo da entrada de diário se RCVSIZOPT (*MAXOPT2 ou *MAXOPT3) for especificado é de 4 000 000 000 bytes. 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 gravações que excedam 15 761 440 bytes, tem de utilizar o suporte *MAXOPT2 ou *MAXOPT3 para impedir a ocorrência de quaisquer erros.

Para obter mais informações sobre o esquema das várias entradas de diário, existe um ficheiro de inclusão em linguagem C, qp0ljrn1.h, incluído no membro QSYSINC/H (QP0LJRNL), que contém detalhes sobre os formatos e conteúdos de dados específicos das entradas de diário do sistema de ficheiros integrado.

Conceitos relacionados

“Ficheiro de dados contínuos” na página 18

Um *ficheiro de dados contínuos* é uma sequência de bytes acessível aleatoriamente, sem qualquer estrutura adicional imposta pelo sistema.

“Directório” na página 4

Um *directório* é um objecto especial 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.

“Ligação simbólica” na página 15

Uma *ligação simbólica*, também conhecida como ligação não permanente, é um nome de caminho contido num ficheiro.

Tarefas relacionadas

“Iniciar registo em diário” na página 107

Execute estes passos para iniciar o registo de alterações de um objecto através do iSeries Navigator.

“Terminar registo em diário” na página 108

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender terminar o registo de alterações neste objecto, pode utilizar os passos descritos neste tópico.

“Alterar registo em diário” na página 107

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender alterar os atributos de diário do objecto sem ter de terminar e reiniciar o registo de alterações, pode utilizar o comando Alterar objecto registado em diário (CHGJRNOBJ), que permite alterar os objectos registados em diário.

Informações relacionadas

Gestão de registos em diário

Localizador de informações de entrada de diário

Operações registadas em diário

Estas operações só são registadas em diário se o tipo de objecto ou ligação que a operação estiver a utilizar também puder ser registado em diário.

- Criar um objecto.
- Adicionar uma ligação a um objecto existente.
- Interromper uma ligação.
- Mudar o nome a uma ligação.
- Mudar o nome a um identificador de ficheiro.
- Mover uma ligação para ou de um directório.

As operações registadas em diário que se seguem são específicas de um ficheiro de dados contínuos:

- Escrita ou eliminação de dados
- Truncagem/extensão de ficheiros
- Dados de ficheiro forçados
- Salvaguarda com memória livre

As operações registadas em diário que se seguem aplicam-se a todos os tipos de objecto registados:

- Alterações aos atributos (incluindo alterações na segurança, tais como autoridades e propriedade)
- Abrir
- Fechar
- Iniciar registo em diário
- Comando Alterar objecto registado em diário (CHGJRNOBJ)
- Terminar registo em diário
- Comando Iniciar a aplicação de alterações registadas em diário (APYJRNCHG)
- Comando Terminar a aplicação de alterações registadas em diário (APYJRNCHG)
- Guardar
- Restaurar

Informações relacionadas

Gestão de registos em diário

Localizador de informações de entrada de diário

Considerações especiais sobre entradas de diário

Muitas operações do sistema de ficheiros integrado registadas em diário utilizam internamente o controlo de consolidações para formar uma única transacção a partir de várias funções executadas durante as operações.

Estas operações registadas em diário não deverão ser consideradas como completas, a menos que o ciclo do controlo de consolidações tenha uma entrada de diário de Consolidações (Código de Diário C, Tipo CM). As operações registadas que contenham uma entrada de diário Remover Alterações (Código C do Registo, Tipo RB) no ciclo de controlo de consolidação correspondem a operações falhadas e as entradas de registo incluídas nas mesmas não devem reproduzidas nem replicadas.

As entradas do sistema de ficheiros integrado registadas em diário (Código de diário B) que utilizam o controlo de consolidações desta forma incluem:

- AA — Alterar valor da auditoria
- B0 — Iniciar criação
- B1 — Criar resumo
- B2 — Adicionar ligação
- B3 — Mudar o nome/Mover
- B4 — Remover ligação (directório ascendente)
- B5 — Remover ligação (Ligação)
- B7 — Informações de autoridade sobre objecto criado
- FA — Alteração de Atributos
- JT — Iniciar Diário [apenas quando o registo de alterações é iniciado devido a uma operação num directório com o atributo de registo de alterações herdado Sim (Yes)]
- 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 uma entrada de resumo. As operações que enviam tipos de entrada de resumo enviam dois dos mesmos tipos de entrada ao registo. A primeira entrada contém um subconjunto dos dados específicos da entrada. A segunda entrada contém dados específicos da entrada completos e indicará que se trata de uma entrada de resumo. Os programas que executam a replicação do objecto ou a reprodução da operação normalmente só se interessam por entradas de resumo.

Para uma operação criar num directório registado em diário, a entrada de registo B1 (Criar Resumo) é considerada como a entrada de resumo.

Algumas operações registadas em diário precisam de enviar uma entrada de registo que esteja de modo oposto relacionada à operação. Por exemplo, um ciclo de controlo de consolidação que contém uma entrada de registo B4 (Desligar) pode também conter uma entrada de registo B2 (Adicionar Ligação). Este tipo de cenário só ocorre em operações que resultam numa entrada de diário Remover Alterações (C — RB).

Ele pode ocorrer por duas razões:

1. A operação estava prestes a falhar e a entrada era necessária internamente para limpeza do caminho com erro.
2. A operação foi interrompida por uma desactivação do sistema e, durante o IPL subsequente, a recuperação que tinha de enviar a entrada foi executada para remover as alterações feitas na operação interrompida.

Informações relacionadas

Localizador de informações de entrada de diário

Considerações sobre múltiplas ligações permanentes e o registo em diário

Se tiver múltiplas ligações permanentes a um objecto do sistema de ficheiros integrado registado em diário, todas as ligações deverão ser guardadas e restauradas ao mesmo tempo para que a ligação seja mantida, bem como, as informações de registo em diário associadas.

Se estiver a especificar nomes em alguns dos comandos relacionados com o registo em diário e se os nomes são na realidade múltiplas ligações permanentes, só será efectuada uma operação com o objecto. As outras ligações permanentes são essencialmente ignoradas.

Uma vez que os vários caminhos directos apontam para o mesmo objecto e a entrada de diário só tem o identificador de ficheiro (ID de ficheiro) que é o mesmo para o objecto, quaisquer interfaces de diário que mostrem o nome de caminho, por exemplo, Apresentar Diário (DSPJRN), apenas mostram um nome de ligação correspondente ao objecto. No entanto, esta situação não deverá causar problemas, uma vez que é possível operar um objecto utilizando qualquer nome e obter os mesmos resultados.

Conceitos relacionados

“Ligação permanente” na página 14

Um *caminho directo*, por vezes, denominado apenas ligação, não pode existir se não estiver ligado a um objecto real.

Iniciar registo em diário

Execute estes passos para iniciar o registo de alterações de um objecto através do iSeries Navigator.

1. Expanda o sistema no **iSeries Navigator**.
2. Expanda **Sistemas de Ficheiros**.
3. Faça clique com o botão direito do rato no objecto cujo registo de alterações pretende efectuar e seleccione **A registar alterações... (Journaling...)**
4. Após seleccionar as opções de registo em diário apropriadas, faça clique sobre **Iniciar**.

Para iniciar o registo de alterações de um objecto através da interface baseada em caracteres, pode utilizar o comando Iniciar diário (STRJRN) ou a API QjoStartJournal.

Conceitos relacionados

“Objectos do sistema de ficheiros integrado registados em diário” na página 103

Alguns tipos de objecto do sistema de ficheiros integrado podem ser registados em diário através da utilização do suporte de registo de alterações do i5/OS.

Tarefas relacionadas

“Alterar registo em diário”

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender alterar os atributos de diário do objecto sem ter de terminar e reiniciar o registo de alterações, pode utilizar o comando Alterar objecto registado em diário (CHGJRNOBJ), que permite alterar os objectos registados em diário.

“Terminar registo em diário” na página 108

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender terminar o registo de alterações neste objecto, pode utilizar os passos descritos neste tópico.

Informações relacionadas

Comando Iniciar diário (STRJRN)

API QjoStartJournal

Gestão de registos em diário

Alterar registo em diário

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender alterar os atributos de diário do objecto sem ter de terminar e reiniciar o registo de alterações, pode utilizar o comando Alterar objecto registado em diário (CHGJRNOBJ), que permite alterar os objectos registados em diário.

Conceitos relacionados

“Objectos do sistema de ficheiros integrado registados em diário” na página 103

Alguns tipos de objecto do sistema de ficheiros integrado podem ser registados em diário através da utilização do suporte de registo de alterações do i5/OS.

Tarefas relacionadas

“Iniciar registo em diário” na página 107

Execute estes passos para iniciar o registo de alterações de um objecto através do iSeries Navigator.

“Terminar registo em diário”

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender terminar o registo de alterações neste objecto, pode utilizar os passos descritos neste tópico.

Informações relacionadas

Comando Alterar objecto registado em diário (CHGJRNOBJ)

Terminar registo em diário

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender terminar o registo de alterações neste objecto, pode utilizar os passos descritos neste tópico.

Para terminar o registo de alterações num objecto através do iSeries Navigator, execute os seguintes passos:

1. Expanda o sistema no **iSeries Navigator**.
2. Expanda **Sistemas de ficheiros (File Systems)**.
3. Faça clique com o botão direito do rato no objecto cujo registo de alterações pretende parar e seleccione **A registar alterações... (Journaling...)**
4. Faça clique em **Terminar (End)**.

Para terminar o registo de alterações de um objecto através da interface baseada em caracteres, pode utilizar o comando Terminar diário (ENDJRN) ou a API QjoEndJournal.

Conceitos relacionados

“Objectos do sistema de ficheiros integrado registados em diário” na página 103

Alguns tipos de objecto do sistema de ficheiros integrado podem ser registados em diário através da utilização do suporte de registo de alterações do i5/OS.

Tarefas relacionadas

“Alterar registo em diário” na página 107

Após o registo de alterações ter sido iniciado num determinado objecto e, por qualquer motivo, pretender alterar os atributos de diário do objecto sem ter de terminar e reiniciar o registo de alterações, pode utilizar o comando Alterar objecto registado em diário (CHGJRNOBJ), que permite alterar os objectos registados em diário.

“Iniciar registo em diário” na página 107

Execute estes passos para iniciar o registo de alterações de um objecto através do iSeries Navigator.

Informações relacionadas

Comando Terminar diário (ENDJRN)

API QjoEndJournal

Gestão de registos em diário

| Regenerar sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador

| A regeneração de sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador pode ser efectuada utilizando os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG).

| Ao utilizar os comandos RCLLNK e RCLSTG, pode executar as seguintes tarefas:

- | • Corrigir problemas do perfil do utilizador do objecto
- | • Corrigir problemas do sistema de ficheiros definido pelo utilizador

- Corrigir problemas de objectos internos
- Remover ligações de objecto inválidas
- Processar objectos danificados
- Criar objectos de sistema em falta
- Corrigir problemas do sistema de ficheiros interno (apenas RCLSTG)
- Localizar objectos perdidos (apenas RCLSTG)

Referências relacionadas

“Sugestões: Regenerar objectos do sistema de ficheiros integrado” na página 101

Enquanto o sistema estiver a converter os sistemas de ficheiros "raiz" (/), QOpenSys e UDFS de ASP do utilizador para suportarem o formato de directório *TYPE2, os comandos Regenerar armazenamento (RCLSTG) e Regenerar ligações de objectos (RCLLNK) não podem ser executados nos directórios do sistema de ficheiros integrado, incluindo os directórios pertencentes às ASPs independentes.

Comparação entre os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG)

Pode utilizar os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG) para corrigir problemas em sistemas de ficheiros "raiz" (/), QOpenSys e definidos pelo utilizador.

O comando RCLLNK identifica e, se possível, corrige problemas em sistemas de ficheiros montados em utilização. O comando RCLSTG não tem esta função. No entanto, o comando RCLSTG permite corrigir problemas que o comando RCLLNK não consegue identificar nem corrigir. A tabela que se segue fornece uma comparação mais detalhada entre os dois comandos.

Tabela 11. Comparação entre o comando RCLLNK e RCLSTG

	RCLLNK OBJ('/MyDir/MyObj')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
É necessário que o sistema se encontre num estado restrito?	Não	Sim	Não
Todos os sistemas de ficheiros estão em condições de ser utilizados durante a operação de regeneração?	Sim	Não	Os sistemas de ficheiros em ASP independente a serem regenerados não estão em condições de ser utilizados.
Em ASPs os objectos podem ser regenerados?	Regenere objectos no sistema, utilizador e ASPs independentes.	Regenere objectos no sistema e ASPs do utilizador.	Regenere objectos em ASPs independentes.
De que forma são regenerados os objectos?	Os objectos são regenerados individualmente ou numa base de subárvore conforme especificado no comando.	Os objectos são regenerados numa base a nível geral do sistema.	Os objectos são regenerados numa base de ASPs independentes.
Quais os problemas conhecidos e aplicáveis do sistema de ficheiros que são identificados e corrigidos, se possível?	A maior parte (Consulte o tópico Regenerar o sistema de ficheiros "raiz" (/), QOpenSys e definido pelo utilizador para obter mais informações.)	Todos	Todos
Os objectos perdidos são localizados?	Não	Sim	Sim

Tabela 11. Comparação entre o comando RCLLNK e RCLSTG (continuação)

	RCLLNK OBJ('/MyDir/MyObj')	RCLSTG ASPDEV(*SYSBAS)	RCLSTG ASPDEV(<IASPNAME>)
Os objectos em sistemas de ficheiros não montados são regenerados?	Não	Sim	Sim
O comando está protegido por módulos?	Sim	Não	Não
Quantas instâncias do comando podem ser executadas simultaneamente?	Várias instâncias	Uma única instância	Uma única instância
Quais os objectos fornecidos pelo sistema de ficheiros integrado que são recriados, se necessário?	Todos	A maior parte (Consulte o tópico Recriar objectos fornecidos pelo sistema de ficheiros integrado para obter mais informações.)	Nenhum
Os objectos danificados podem se identificados sem serem regenerados?	Sim	Não	Não

Conceitos relacionados

“Exemplos: comando Regenerar ligações de objectos (RCLLNK)” na página 111

Estes exemplos descrevem situações em que o comando Regenerar ligações de objectos (RCLLNK) pode ser utilizado para regenerar objectos em sistemas de ficheiros “raiz” (/), QOpenSys e montados definidos pelo utilizador.

Referências relacionadas

“Recriar objectos fornecidos pelo sistema de ficheiros integrado” na página 111

Esta tabela mostra os objectos fornecidos pelo sistema de ficheiros integrado recriados pelo comando Regenerar ligações de objectos (RCLLNK), caso não existam. Estes objectos são criados geralmente durante o carregamento do programa inicial (IPL - Initial Program Load). Se necessário, também pode recriar alguns destes objectos utilizando o comando Regenerar armazenamento (RCLSTG).

Informações relacionadas

Comando Regenerar armazenamento (RCLSTG)

Comando Regenerar ligações de objectos (RCLLNK)

Comando Regenerar ligações de objectos (RCLLNK)

O comando Regenerar ligações de objectos (RCLLNK) identifica e repara objectos danificados nos sistemas de ficheiros “raiz” (/), QOpenSys e montados definidos pelo utilizador sem ser requerido que o sistema esteja no estado restrito. Deste modo, pode corrigir problemas nestes sistemas de ficheiros sem comprometer a produtividade.

O comando RCLLNK pode ser utilizado como alternativa ao comando Regenerar armazenamento (RCLSTG) em várias situações. Por exemplo, RCLLNK é ideal para identificar e corrigir problemas nas seguintes situações:

- Problemas isolados num único objecto.
- Problemas isolados num grupo de objectos.
- Objectos danificados que necessitem ser identificados ou eliminados.
- O sistema não pode estar no estado restrito durante a operação de regeneração.
- As ASPs independentes têm de estar disponíveis durante a operação de regeneração.

Recriar objectos fornecidos pelo sistema de ficheiros integrado

Esta tabela mostra os objectos fornecidos pelo sistema de ficheiros integrado recriados pelo comando Regenerar ligações de objectos (RCLLNK), caso não existam. Estes objectos são criados geralmente durante o carregamento do programa inicial (IPL - Initial Program Load). Se necessário, também pode recriar alguns destes objectos utilizando o comando Regenerar armazenamento (RCLSTG).

Tabela 12. Objectos fornecidos pelo sistema de ficheiros integrado e recriados pelos comandos RCLLNK e RCLSTG

Nome de caminho	Tipo	Recriados por RCLLNK	Recriados por RCLSTG ASPDEV(*SYSBASE)
/dev/zero	*CHRFS	Sim	Sim
/dev/null	*CHRFS	Sim	Sim
/dev/xti/tcp	*CHRFS	Sim	Não
/dev/xti/udp	*CHRFS	Sim	Não
/etc/vfs	*STMF	Sim	Não

Para que o comando RCLLNK recrie um objecto fornecido pelo sistema de ficheiros integrado que não exista, tem de ser executado com o parâmetro SUBTREE definido como *DIR ou *ALL, sendo especificado o directório ascendente. O comando tem de regenerar com êxito o directório ascendente do objecto de sistema. Por exemplo,

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

recria os objectos /dev/zero e /dev/null *CHRFS, caso não existam.

Para que o comando RCLSTG recrie objecto fornecido pelo sistema de ficheiros integrado que não exista, tem de ser executado com o parâmetro ASPDEV definido como *SYSBASE e a parte de recuperação do directório de regeneração não pode ser omitida.

Conceitos relacionados

“Directórios fornecidos” na página 7

Ao reiniciar o sistema, o sistema de ficheiros integrado cria estes directórios, caso ainda não existam.

Referências relacionadas

“Comparação entre os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG)” na página 109

Pode utilizar os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG) para corrigir problemas em sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador.

Informações relacionadas

Comando Regenerar ligações de objectos (RCLLNK)

Exemplos: comando Regenerar ligações de objectos (RCLLNK)

Estes exemplos descrevem situações em que o comando Regenerar ligações de objectos (RCLLNK) pode ser utilizado para regenerar objectos em sistemas de ficheiros “raiz” (/), QOpenSys e montados definidos pelo utilizador.

Referências relacionadas

“Comparação entre os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG)” na página 109

Pode utilizar os comandos Regenerar ligações de objectos (RCLLNK) e Regenerar armazenamento (RCLSTG) para corrigir problemas em sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador.

| **Exemplo: Corrigir problemas de um objecto**

| Nesta situação, os problemas conhecidos são isolados num único objecto. O objecto está danificado e não está em condições de ser utilizado e o utilizador pretende restaurar uma versão da cópia de segurança do objecto a partir do suporte. Tem de corrigir o problema rapidamente sem interromper as operações normais do sistema de ficheiros.

| Para regenerar o objecto, utilize este comando:

```
| RCLLNK OBJ('/MeuDir/MeuObjectoDanificado') SUBTREE(*NONE)
```

| em que /MeuDir/MeuObjectoDanificado corresponde ao nome do objecto danificado e que não está em condições de ser utilizado.

| **Exemplo: Corrigir problemas que existam na subárvore de um directório**

| Nesta situação, os problemas conhecidos são isolados num grupo de objectos na subárvore de um directório. Uma aplicação está a falhar devido a problemas na subárvore do directório. Tem de corrigir os problemas rapidamente sem interromper as operações normais do sistema de ficheiros.

| Para regenerar os objectos na subárvore do directório, utilize este comando:

```
| RCLLNK OBJ('/MeuDirectórioInstalaçãoAplicação') SUBTREE(*ALL)
```

| em que MeuDirectórioInstalaçãoAplicação corresponde ao nome do directório que contém os objectos com problemas.

| **Exemplo: Localizar todos os objectos danificados em sistemas de ficheiros "raiz" (/), QOpenSys e montados definidos pelo utilizador**

| Nesta situação, um erro de disco causou danos a um determinado número de objectos. Tem de identificar os objectos danificados para determinar como recuperá-los de forma adequada.

| Necessita de uma solução que permita identificar os objectos danificados, mas que não implique que seja executada qualquer acção relativamente aos mesmos. Não pode interromper as operações normais do sistema de ficheiros.

| Para identificar os objectos danificados, utilize este comando:

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

| Além disso, este comando também corrige problemas diferentes de objectos danificados ao identificar objectos danificados.

| **Exemplo: Eliminar todos os objectos danificados em sistemas de ficheiros "raiz" (/), QOpenSys e montados definidos pelo utilizador**

| Nesta situação, um erro de disco fez com que um determinado número de objectos ficasse danificado. Tem de eliminar os objectos danificados para que seja possível restaurar uma cópia de segurança dos objectos a partir do suporte.

| Para eliminar os objectos danificados, utilize este comando:

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

| Os objectos danificados são eliminados sem interrupção das operações normais do sistema de ficheiros. Além disso, os problemas não relacionados com a danificação são corrigidos à medida que os objectos danificados forem eliminados.

| **Exemplo: Executar vários comandos RCLLNK para regenerar rapidamente todos os objectos em sistemas de ficheiros "raiz" (/), QOpenSys e montados definidos pelo utilizador**

| Nesta situação, como parte da manutenção do sistema de rotina, todos os objectos nos sistemas de ficheiros "raiz" (/), QOpenSys e montados definidos pelo utilizador são regenerados. Pretende terminar a operação de regeneração o mais rápido possível para permitir que as informações adicionais do sistema sejam concluídas.

| Ao dividir a operação de regeneração em dois grupos em separado, podem ser executados vários comandos RCLLNK em simultâneo para permitir que a operação de regeneração termine logo que possível.

| Para executar várias operações de regeneração em directórios de sistema chave e noutros directórios de nível superior, utilize os seguintes comandos (cada um num trabalho ou módulo em separado).

```
| RCLLNK OBJ('/') SUBTREE(*DIR)
| RCLLNK OBJ('/tmp') SUBTREE(*ALL)
| RCLLNK OBJ('/home') SUBTREE(*ALL)
| RCLLNK OBJ('/etc') SUBTREE(*ALL)
| RCLLNK OBJ('/utlz') SUBTREE(*ALL)
| RCLLNK OBJ('/QIBM') SUBTREE(*ALL)
| RCLLNK OBJ('/QOpenSys') SUBTREE(*ALL)
| RCLLNK OBJ('/NomeIasp') SUBTREE(*ALL)
| RCLLNK OBJ('/dev') SUBTREE(*ALL)
| RCLLNK OBJ('/OutrosDirectóriosNívelSuperior') SUBTREE(*ALL)
```

| em que OutrosDirectóriosNívelSuperior corresponde a outros directórios que pretenda regenerar.

Suporte de programação

A adição do sistema de ficheiros integrado ao servidor servidor iSeries na versão V3R1M0 não afectou as aplicações do servidor iSeries existentes. As linguagens de programação, utilitários e suporte do sistema (por exemplo, especificações de descrição de dados), funcionam do mesmo modo que funcionavam antes da adição do sistema de ficheiros integrado.

No entanto, para tirar partido dos ficheiros de dados contínuos, directórios e outros suportes do sistema de ficheiros integrado, tem de utilizar um conjunto de interfaces de programas de aplicações (API - Application Program Interface) fornecido para aceder a funções do sistema de ficheiros integrado.

Além disso, a adição do sistema de ficheiros integrado permite 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 Família iSeries Access ou APIs.

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 utilizando serviços orientados para os registos, tais como especificações de descrição de dados (DDS - Data Description Specification), 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.

Conceitos relacionados

“Ficheiro de dados contínuos” na página 18

Um *ficheiro de dados contínuos* é uma sequência de bytes acessível aleatoriamente, sem qualquer estrutura adicional imposta pelo sistema.

Copiar dados utilizando comandos de CL

Existem dois conjuntos de comandos de CL que permitem copiar dados entre ficheiros de dados contínuos e membros de ficheiros de base de dados.

Comandos CPYTOSTMF e CPYFRMSTMF

Pode utilizar os comandos Copiar a partir do ficheiro de dados contínuos (CPYFRMSTMF) e Copiar para ficheiro de dados contínuos (CPYTOSTMF) para copiar dados entre ficheiros de dados contínuos e membros do ficheiro de base de dados. Pode criar um ficheiro de dados contínuos a partir de um membro do ficheiro de base de dados utilizando o comando CPYTOSTMF. Também pode criar um membro do 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 também podem ser utilizados para copiar dados entre um ficheiro de dados contínuos e um ficheiro de salvaguarda.

Comandos CPYTOIMPF e CPYFRMIMPF

Também pode utilizar os comandos Copiar para ficheiro de importação (CPYTOIMPF) e Copiar a partir 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 permitem mover dados de ficheiros de base de dados complexos e descritos externamente (descritos em DDS). A expressão *ficheiro de importação* refere-se ao ficheiro do tipo dados contínuos; o termo refere-se normalmente a um ficheiro criado com a finalidade de copiar dados entre bases de dados heterogéneas.

Ao copiar a partir de um ficheiro de dados contínuos (ou de importação), o comando CPYFRMIMPF permite especificar um ficheiro de definição de campo (FDF - Field Definition File), 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(*DBLQUOTE) DATFMT(*USA) TIMFMT(*USA)
```

O parâmetro DTAFMT especifica que o ficheiro de dados contínuos (importação) de entrada é delimitado; a outra opção é DTAFMT(*FIXED), que requer a especificação de um ficheiro de definição de campo. Os parâmetros FLDDLML, RCDDLML e STRDLML identificam os caracteres que funcionam como delimitadores ou separadores dos campos, registos e cadeias.

Os parâmetros DATFMT e TIMFMT indicam o formato das informações de data e hora que são copiadas para o ficheiro de importação.

Os comandos são úteis porque podem ser colocados num programa, sendo executados inteiramente no servidor. No entanto, as interfaces são complexas.

Informações relacionadas

Comando Copiar para ficheiro de dados contínuos (CPYTOSTMF)

Comando Copiar a partir do ficheiro de dados contínuos (CPYFRMSTMF)

Comando Copiar para ficheiro de importação (CPYTOIMPF)

Comando Copiar a partir de ficheiro de importação (CPYFRMIMPF)

CL (Control language)

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 no mesmo ou noutra ficheiro.

Referências relacionadas

“Executar operações utilizando APIs” na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Informações relacionadas

open()

read()

write()

APIs do sistema de ficheiros integrado

Copiar dados utilizando funções de transferência de dados

As aplicações de transferência de dados existentes no programa licenciado Família iSeries Access apresentam como vantagens uma interface gráfica acessível, bem como a conversão automática de dados em caracteres e valores numéricos.

No entanto, a transferência de dados requer a instalação do produto Família iSeries Access e a utilização de recursos de PC e do servidor iSeries e comunicações entre os dois.

Se tiver instalado o Família iSeries Access no PC e no seu servidor, pode utilizar as aplicações de transferência de dados para transferir dados entre ficheiros de dados contínuos e ficheiros de base de dados. Pode também transferir dados para um novo ficheiro de base de dados, com base num ficheiro de base de dados existente; para um ficheiro de base de dados descrito externamente ou para um novo ficheiro e uma nova definição de ficheiro de base de dados.

Transferir dados de um ficheiro de base de dados para um ficheiro de dados contínuos:

Siga estas instruções para transferir um ficheiro de base de dados para um ficheiro de dados contínuos 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. Na janela do iSeries Access for Windows, faça clique em **Transferência de dados a partir do servidor iSeries** (Data Transfer from iSeries server).
4. Selecciono o servidor a partir do qual pretende efectuar a transferência.
5. Selecciono 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 em **Transferir dados do iSeries (Transfer data from 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 seccione a opção de menu **Ficheiro (File)** para guardar o pedido de transferência. A aplicação Transferência de Dados para o servidor iSeries cria um ficheiro .DTT ou .TFR. A aplicação Transferência de Dados do servidor iSeries cria um ficheiro .DTF ou .TTO. No directório Família iSeries Access, podem ser executados dois programas em batch a partir da linha de comandos:

- RTOPCB requer um ficheiro .DTF ou .TTO como parâmetro
- RFROMPCB requer um ficheiro .DTT ou .TFR como parâmetro

Pode definir qualquer um destes comandos para execução programada utilizando uma aplicação de marcação. Por exemplo, pode utilizar a ferramenta System Agent Tool (incluída no pacote Microsoft® Plus) para especificar o programa a executar (por exemplo, RTOPCB MYFILE.TT0) e a hora a que pretende executar o programa.

Transferir dados de um ficheiro de dados contínuos para um ficheiro de base de dados:

Siga estas instruções para transferir dados a partir 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. Na janela do iSeries Access for Windows, faça clique em **Transferência de dados para o servidor iSeries (Data Transfer to iSeries Server)**.
4. Selecciono o nome do ficheiro de PC que pretende transferir. Para o nome do ficheiro de PC, pode seleccionar **Procurar (Browse)** 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. Selecciono o servidor onde pretende que o ficheiro de base de dados descrito externamente fique localizado.
6. Faça clique em **Transferir dados para o servidor iSeries (Transfer data to servidor 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 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 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 em **Transferir dados para o iSeries (Transfer data to iSeries server)**. Se não estiver disponível um ficheiro .FDF existente, pode criar rapidamente 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 (File)** para guardar o pedido de transferência. A aplicação Transferência de Dados para o servidor iSeries cria um ficheiro .DTT ou .TFR. A aplicação Transferência de Dados do servidor iSeries cria um ficheiro .DTF ou .TTO. No directório Família IBM eServer™ iSeries Access, podem ser executados dois programas em batch a partir da linha de comandos:

- RTOPCB requer um ficheiro .DTF ou .TTO como parâmetro
- RFROMPCB requer um ficheiro .DTT ou .TFR como parâmetro

Pode definir qualquer um destes comandos para execução programada utilizando uma aplicação de marcação. Por exemplo, pode utilizar a ferramenta System Agent Tool (incluída no pacote Microsoft® Plus) para especificar o programa a executar (por exemplo, RTOPCB MYFILE.TT0) e a hora a que pretende executar o programa.

Referências relacionadas

“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 iSeries (Data Transfer to iSeries server) requer a utilização de um ficheiro de descrição de formato (FDF - Format Description File) associado.

Transferir dados para uma definição de ficheiro e ficheiro de base de dados recentemente criado:

Sida estas indicações transferir dados para um ficheiro e definição de 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. Na janela do iSeries Access for Windows, faça clique em **Transferência de dados para o servidor iSeries** (Data Transfer to iSeries server).
4. Abra Ferramentas (Tools) na aplicação Transferência de dados para o servidor iSeries (Data Transfer to iSeries server).
5. Faça clique em **Criar ficheiro de base de dados do iSeries (Create iSeries database file)**.

É apresentado um assistente que permite criar um novo ficheiro de base de dados do iSeries a partir de um ficheiro de PC existente. Tem de 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 determinado ficheiro de dados contínuos 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 iSeries (Data Transfer to iSeries server) requer a utilização de um ficheiro de descrição de formato (FDF - Format Description File) associado.

Um ficheiro FDF descreve o formato de um ficheiro de dados contínuos e é criado pela aplicação Transferência de Dados do servidor 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 iSeries (Data Transfer from iSeries server) para criar um ficheiro de dados contínuos e respectivo ficheiro .FDF associado a partir deste ficheiro de base de dados.

A partir deste momento, já pode utilizar a função Transferência de Dados para o servidor iSeries (Data Transfer to iSeries server). Especifique este ficheiro .FDF com o ficheiro de dados contínuos origem que pretende transferir.

Referências relacionadas

“Transferir dados de um ficheiro de dados contínuos para um ficheiro de base de dados” na página 116

Siga estas instruções para transferir dados a partir de um ficheiro de dados contínuos para um ficheiro de base de dados no servidor.

“Transferir dados de um ficheiro de base de dados para um ficheiro de dados contínuos” na página 116

Siga estas instruções para transferir um ficheiro de base de dados para um ficheiro de dados contínuos no servidor.

Copiar dados entre ficheiros de dados contínuos e ficheiros de salvaguarda

É utilizado um ficheiro de salvaguarda com comandos de salvaguarda e restauro para reter dados que, de outra forma, seriam gravados 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 Copiar objecto (CPY) para copiar um ficheiro de salvaguarda de e para 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 Network File System (NFS).

Informações relacionadas

Comando Copiar objecto (CPY)

Executar operações utilizando APIs

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se 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 i5/OS.
- Funções C fornecidas pelo programa licenciado ILE C/400.

Para obter informações sobre os programas de saída suportados pelo sistema de ficheiros integrado, consulte a Tabela 14 na página 124.

As funções do sistema de ficheiros integrado operam apenas através do suporte de E/S de sequência do sistema de ficheiros integrado. São suportadas as seguintes APIs:

Tabela 13. 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
fclear()	Eliminar um ficheiro
fclear64()	Eliminar um ficheiro (suporte de ficheiros grandes)
fcntl()	Executar acção de controlo de ficheiros
fpathconf()	Obter variáveis do nome de caminho configuráveis por descritor
fstat()	Obter informações sobre o ficheiro por descritor
fstat64()	Obter informações sobre o ficheiro por descritor (suporte de ficheiros grandes)
fstatvfs()	Obter informações por descritor
fstatvfs64()	Obter informações por descritor (64 bits activado)
fsync()	Sincronizar alterações ao ficheiro
ftruncate()	Truncar ficheiro
ftruncate64()	Truncar ficheiro (suporte de ficheiros grandes)
getcwd()	Obter nome de caminho do directório actual
getegid()	Obter ID do grupo efectivo
geteuid()	Obter ID do utilizador efectivo
getgid()	Obter ID de grupo real
getrgid()	Obter informações sobre o grupo utilizando o ID do grupo

Tabela 13. APIs do sistema de ficheiros integrado (continuação)

Função	Descrição
getgrnam()	Obter informações sobre o grupo utilizando o nome do grupo
getgroups()	Obter IDs do grupo
getpwnam()	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
pwrite64()	Escrever no descritor com deslocamento (suporte de ficheiros grandes)
QjoEndJournal()	Terminar registo em diário
QjoRetrieveJournalEntries()	Obter Entradas de Diário
QjoRetrieveJournal Information()	Obter informações do registo em diário
QJORJIDI()	Obter informações do identificador do registo em diário
QJOSJRNE()	Enviar entrada de registo em diário
QjoStartJournal()	Iniciar registo em diário

Tabela 13. APIs do sistema de ficheiros integrado (continuação)

Função	Descrição
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 de caminho do directório actual (utilizando nome de caminho activado para NLS)
QlgGetPathFromFileID()	Obter nome de caminho do objecto a partir do respectivo identificador de ficheiro (utilizando nome de caminho activado para NLS)
QlgGetpwnam()	Obter informações sobre o utilizador para o nome do utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwnam_r()	Obter informações sobre o utilizador para o nome do utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwuid()	Obter informações sobre o utilizador para o ID de utilizador (utilizando nome de caminho activado para NLS)
QlgGetpwuid_r()	Obter informações sobre o utilizador para o ID de utilizador (utilizando nome de caminho activado para NLS)
QlgLchown()	Alterar proprietário e grupo da ligação simbólica (utilizando nome de caminho activado para NLS)
QlgLink()	Criar ligação ao ficheiro (utilizando nome de caminho activado para NLS)
QlgLstat()	Obter informações sobre o ficheiro ou a ligação (utilizando nome de caminho activado para NLS)
QlgLstat64()	Obter informações sobre o ficheiro ou a ligação (suporte para ficheiros grandes e nome de caminho activado para NLS)
QlgMkdir()	Criar directório (utilizando nome de caminho activado para NLS)
QlgMkfifo()	Criar ficheiro especial FIFO (utilizando nome de caminho activado para NLS)
QlgOpen()	Abrir ficheiro (utilizando nome de caminho activado para NLS)

Tabela 13. APIs do sistema de ficheiros integrado (continuação)

Função	Descrição
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
QP0LCHSG()	Alterar assinatura de pesquisa
Qp0lCvtPathToSYSObjName()	Processar nome de caminho do sistema de ficheiros integrado para Nome de Objecto de QSYS
QP0LFLOP()	Executar operações diversas em objectos
Qp0lGetAttr()	Obter atributos de sistema para um objecto
Qp0lGetPathFromFileID()	Obter nome de caminho do objecto a partir do respectivo ID do ficheiro

Tabela 13. APIs do sistema de ficheiros integrado (continuação)

Função	Descrição
Qp0lOpen()	Abrir ficheiro com o nome de caminho activado para NLS
Qp0lProcessSubtree()	Processar directórios ou objectos numa árvore de directórios
Qp0lRenameKeep()	Mudar o nome do ficheiro ou directório, manter <i>novo</i> , se existir
Qp0lRenameUnlink()	Mudar o nome do ficheiro ou directório, desligar <i>novo</i> se existir
QP0LROR()	Obter referências a objectos
QP0LRRO()	Obter objectos referenciados
QP0LRTSG()	Obter assinatura de pesquisa
Qp0lSaveStgFree()	Guardar dados de objectos e libertar a respectiva memória
Qp0lSetAttr()	Definir atributos de sistema para um objecto
Qp0lUnlink()	Desligar ficheiro com o nome de caminho activado para NLS
qsysetegid()	Definir ID de grupo efectivo
qsyseteuid()	Definir ID de utilizador efectivo
qsysetgid()	Definir ID de grupo
qsysetregid()	Definir IDs de grupo reais e efectivos
qsysetreuid()	Definir IDs de utilizador reais e efectivos
qsysetuid()	Definir ID de utilizador
QZNFRTVE()	Obter informações de exportação de NFS
read()	Ler do ficheiro
readdir()	Ler entrada de directório
readdir_r()	Ler entrada de directório (protecção por módulo)
readlink()	Ler valor da ligação simbólica
readv()	Ler do ficheiro (vector)
rename()	Mudar o nome do ficheiro ou ao directório. Pode ser definido para ter a semântica Qp0lRenameKeep() ou Qp0lRenameUnlink().
rewinddir()	Repor sequência do directório
rmdir()	Remover directório
select()	Verificar estado de I/O de vários descritores de ficheiros
stat()	Obter informações sobre o ficheiro
stat64()	Obter informações sobre o ficheiro (suporte de ficheiros grandes)
statvfs()	Obter informações sobre o sistema de ficheiros
statvfs64()	Obter informações sobre o sistema de ficheiros (suporte de ficheiros grandes)
symlink()	Criar ligação simbólica
sysconf()	Obter variáveis de configuração do sistema
takedescriptor()	Obter acesso ao ficheiro a partir de outro trabalho
umask()	Definir máscara de autorização para o trabalho
unlink()	Remover ligação ao ficheiro
utime()	Definir horas de acesso e de modificação do ficheiro

Tabela 13. APIs do sistema de ficheiros integrado (continuação)

Função	Descrição
write()	Escrever no ficheiro
writenv()	Escrever no ficheiro (vector)

Nota: Algumas destas funções também são utilizadas para terminais do i5/OS.

Tabela 14. Programas de saída do sistema de ficheiros integrado

Função	Descrição
Pesquisa do Sistema de Ficheiros Integrado em API Fechada	Chamada durante o processamento de encerramento, tal como com a API close(). Este programa de saída tem de ser fornecido pelo utilizador.
Pesquisa do Sistema de Ficheiros Integrado em API Aberta	Chamada durante o processamento de abertura, tal como com a API open(). Este programa de saída tem de ser fornecido pelo utilizador.
Processar um Nome de Caminho	Chamado pela API Qp0IProcessSubtree() para todos os objectos na procura da API que correspondem aos critérios de selecção do programa de chamada. Este programa de saída tem de ser fornecido pelo utilizador.
Guardar Libertação de Memória	Chamado pela API Qp0ISaveStgFree() para guardar um tipo de objecto *STMF iSeries. Este programa de saída tem de ser fornecido pelo utilizador.

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Referências relacionadas

“Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros “raiz” (/)” na página 35
Todas as APIs listadas no tópico “Executar operações utilizando APIs” podem operar no sistema de ficheiros “raiz” (/).

“Utilizar APIs do sistema de ficheiros integrado no sistema de ficheiros definido pelo utilizador” na página 42

Todas as APIs listadas no tópico “Executar operações utilizando APIs” podem operar num sistema de ficheiros definido pelo utilizador.

“Aceder utilizando APIs” na página 93

Pode utilizar interfaces do programas de aplicações (API - Application Program Interfaces) para aceder ao sistema de ficheiros integrado.

“Exemplo: Funções C do sistema de ficheiros integrado” na página 130

Este simples programa em linguagem C ilustra a utilização de várias funções do sistema de ficheiros integrado.

“Copiar dados utilizando APIs” na página 115

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 no mesmo ou noutra ficheiro.

Informações relacionadas

Interfaces de programação de aplicações (APIs)

Funções de ILE C/400

O ILE C/400 fornece as funções C padrão definidas pelo American National Standards Institute (ANSI).

Estas funções podem funcionar através do suporte de E/S de gestão de dados ou através do suporte de E/S de sequência do sistema de ficheiros integrado, dependendo das informações especificadas ao 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 a E/S de sequência do sistema de ficheiros integrado, tem de especificar *IFSIO no 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 E/S do sistema de ficheiros integrado são ligadas em vez das funções de E/S de gestão de dados. Com efeito, as funções C de ILE C/400 utilizam as funções do sistema de ficheiros integrado para executar a E/S.

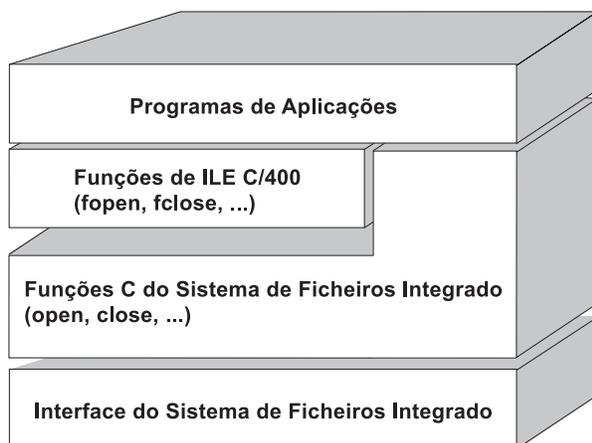


Figura 10. As funções de ILE C/400 utilizam as funções de E/S de sequência do sistema de ficheiros integrado.

Para obter mais informações sobre como utilizar as funções de ILE C/400 com a E/S de sequência do sistema de ficheiros integrado, consulte a publicação WebSphere® Development Studio: Manual dos programadores de ILE C/C++ . Para obter detalhes sobre cada função C de ILE C/400, consulte a publicação WebSphere Development Studio: Referência de Linguagem C/C++ .

Suporte de ficheiros grandes

As APIs do sistema de ficheiros integrado foram melhoradas para permitir que as aplicações armazenem e manipulem ficheiros muito grandes. O sistema de ficheiros integrado permite tamanhos de ficheiros de dados contínuos até aproximadamente 1 TB (1 TB equivale a aproximadamente 1 099 511 627 776 bytes) nos sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador.

O sistema de ficheiros integrado fornece um conjunto de APIs de tipo UNIX de 64 bits e permite um mapeamento fácil de APIs de 32 bits existentes para APIs de 64 bits com capacidade para aceder a deslocamentos e ficheiros de tamanhos grandes através da utilização de argumentos de número inteiro de oito bytes.

The following situations are provided to allow applications to use large file support:

- Se a etiqueta de macro `_LARGE_FILE_API` estiver definida no momento da compilação, as aplicações terão acesso a APIs e a estruturas de dados que suportam 64 bits. Por exemplo, uma aplicação que tencione utilizar a API `stat64()` e a estrutura `stat64` terá de definir `_LARGE_FILE_API` no momento da compilação.
- 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` durante o tempo de compilação, uma chamada para a API `stat()` é mapeada para a API `stat64()` e a estrutura `stat()` é mapeada para 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 tenham como finalidade utilizar o suporte de ficheiros grandes não são afectadas e podem continuar a utilizar as APIs do sistema de ficheiros integrado sem quaisquer alterações.

Informações relacionadas

APIs do sistema de ficheiros integrado

`stat64()`

`stat()`

Regras de nomes de caminhos para APIs

Quando utilizar uma API de ILE C/400[®] ou do sistema de ficheiros integrado para operar um objecto, o objecto é identificado fornecendo o respectivo caminho do directório. Segue-se um resumo das regras a ter em consideração ao especificar nomes de caminhos 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.
- Uma / no início de um nome de caminho significa que o caminho começa no directório "raiz" (/); por exemplo:

```
/Dir1/Dir2/Dir3/FichUtil
```

- Se o nome de caminho não começar por /, assume-se que o caminho começa no directório actual; por exemplo:

```
MeuDir/MeuFich
```

onde MeuDir é um subdirectório do directório actual.

- Para evitar uma possível confusão com valores especiais do servidor 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 operar 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
```

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

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

- Não utilize dois pontos (:) em nomes de caminho. Têm um significado especial no servidor.
- Ao contrário do que acontece com os nomes de caminhos nos comandos do sistema de ficheiros integrado, um asterisco (*), um ponto de interrogação (?), uma plica ('), as aspas (") e um til (~) não têm qualquer significado especial. São processados como qualquer outro carácter no nome. Para evitar

uma possível confusão com valores especiais do servidor iSeries, os nomes de caminhos não devem ser iniciados por um único carácter de asterisco (*). As únicas APIs que se apresentam como excepções a esta regra são QjoEndJournal e QjoStartJournal.

- Quando utilizar as interfaces API Qlg (utilizando nomes de caminhos activados para NLS), não é permitido um valor de carácter nulo como um dos caracteres do nome de caminho, excepto se for especificado um carácter nulo como um delimitador do nome de caminho.

Conceitos relacionados

“Nome de caminho” na página 16

Um *nome de caminho*) indica ao servidor como localizar um objecto.

Referências relacionadas

“Regras de nomes de caminhos para comandos de CL e ecrãs” na página 77

Ao utilizar um comando ou ecrã do sistema de ficheiros integrado para operar um objecto, este é identificado fornecendo o respectivo nome de caminho.

Informações relacionadas

API QjoEndJournal

API 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. Ao utilizar as funções C do sistema de ficheiros integrado, identifique o ficheiro especificando um descritor do ficheiro. Um *descritor do ficheiro* é um número inteiro positivo que tem de 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 do ficheiro é representado pela variável *files* nas funções C que operam no sistema de ficheiros integrado e pela variável *descriptor* nas funções C que operam nos terminais.

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.

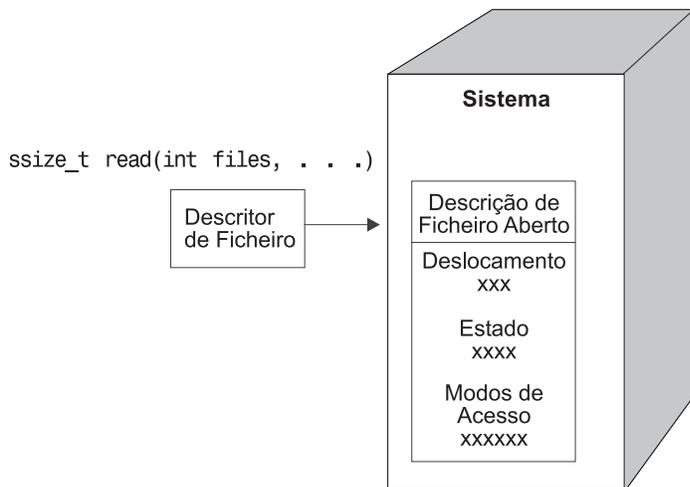


Figura 11. Descritor de ficheiros e descrição de ficheiro aberto

Se uma função de E/S de sequência ILE C/400 for utilizada com o sistema de ficheiros integrado, o suporte de tempo de execução de ILE C/400 converte o apontador de ficheiro num descritor do ficheiro.

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys ou definidos pelo utilizador, pode transferir o acesso a uma descrição de ficheiro aberto de um trabalho para outro, permitindo, deste modo, que o trabalho tenha acesso ao ficheiro. Para tal, utilize a função `givedescriptor()` ou `takedescriptor()` para transferir o descritor de ficheiros entre trabalhos.

Informações relacionadas

API `givedescriptor()`

API `takedescriptor()`

Programação de terminais

APIs de terminal

Segurança

Quando utilizar as APIs do sistema de ficheiros integrado, pode restringir o acesso a objectos, tal como quando utiliza interfaces de gestão de dados. No entanto, tenha em consideração que a adopção de autoridades não é suportada. Uma API do sistema de ficheiros integrado utiliza a autoridade do perfil de utilizador no qual o trabalho estiver 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 Network File System são executados no perfil do utilizador cujo número de identificação de utilizador (UID) foi recebido pelo servidor de NFS no momento do pedido.

As autoridades no servidor são o equivalente às *permissões* nos sistemas UNIX. Os tipos de permissões são leitura e escrita (para um ficheiro ou directório) e executar (para um ficheiro) ou procurar (para um directório). As permissões são indicadas por um conjunto de bits de permissão, que constituem o “modo de acesso” do ficheiro ou directório. Pode alterar os bits de permissão utilizando as funções “alterar modo” `chmod()` ou `fchmod()`. Também pode utilizar a função `umask()` para controlar quais os bits de permissão de ficheiro definidos sempre que um trabalho cria um ficheiro.

Para obter detalhes sobre segurança de dados e autoridades, consulte a publicação Security Reference



Informações relacionadas

API `chmod()`

API `fchmod()`

API `umask()`

APIs do sistema de ficheiros integrado

Suporte de sockets

Se a aplicação estiver a utilizar sistemas de ficheiros “raiz” (/), QOpenSys ou definidos pelo utilizador pode tirar partido do suporte de *terminal local* do sistema de ficheiros integrado. Um objecto socket local (objecto do tipo *SOCKET) permite a execução de dois trabalhos no mesmo sistema, de forma a estabelecer uma ligação de comunicações entre eles.

Um dos trabalhos estabelece um ponto de ligação utilizando a função de linguagem C `bind()` para criar o objecto de terminal local. O outro trabalho especifica o nome do objecto de terminal local na função `connect()`, `sendto()` ou `sendmsg()`.

Após estabelecer a ligação, os dois trabalhos podem enviar e receber dados entre si utilizando funções do sistema de ficheiros integrado, tais como 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 encerrar a ligação de terminais. O objecto de terminal local permanece no sistema até ser removido com a função unlink() ou com o comando Remover ligação (RMVLNK).

O objecto socket local não pode ser guardado.

Informações relacionadas

Programação de terminais

API write()

API read()

API close()

API unlink()

Comando Remover ligação (RMVLNK)

Nomenclatura e suporte internacional

O suporte dos sistemas de ficheiros “raiz” (/) e QOpenSys garante que os caracteres nos nomes de objectos permaneçam constantes nos esquemas de codificação utilizados para diferentes idiomas nacionais e dispositivos.

Quando um nome de objecto é transferido para o sistema, cada carácter do nome é convertido num formato de 16 bits, em que todos os caracteres têm uma representação codificada padrão. Quando é utilizado, o nome é convertido no formato codificado apropriado para a página de códigos a ser utilizada.

Se a página de códigos para a qual o nome está a ser convertido não contiver um carácter utilizado num nome, o nome é recusado como sendo inválido.

Uma vez que os caracteres permanecem constantes entre diversas páginas de códigos, não deve efectuar uma operação pressupondo que um determinado carácter será alterado para outro carácter quando é utilizada uma página de códigos específica. Por exemplo, não deverá partir do princípio que o carácter cardinal (#) mudará para o carácter de libra esterlina, mesmo que possam ter a mesma representação codificada em páginas de códigos diferentes.

Repare que os nomes dos atributos expandidos de um objecto são convertidos do mesmo modo que o nome do objecto, por conseguinte, também se aplicam as mesmas considerações.

Conceitos relacionados

“Continuidade de nomes” na página 19

Ao utilizar os sistemas de ficheiros “raiz” (/), QOpenSys e definidos pelo utilizador, pode tirar partido do suporte de sistemas que permite garantir que os caracteres nos nomes de objectos permanecem iguais.

Conversão de dados

Ao aceder a ficheiros através do sistema de ficheiros integrado, os dados dos ficheiros podem ou não ser convertidos, dependendo do modo de abertura solicitado ao abrir o ficheiro.

Um ficheiro aberto pode estar num ou em dois modos de abertura:

Binário

Os dados são lidos do ficheiro e escritos para o ficheiro sem conversão. A aplicação é responsável pelo tratamento dos dados.

Texto Os dados são lidos do ficheiro e escritos no ficheiro, assumindo que se encontram em formato de texto. Quando os dados são lidos do ficheiro, são convertidos do CCSID (coded character set identifier, identificador do conjunto de caracteres codificados) do ficheiro para o CCSID da aplicação, do trabalho ou do sistema que recebe os dados. Quando os dados são escritos no ficheiro, são convertidos do CCSID da aplicação, do trabalho ou do sistema para o CCSID do ficheiro. Para os verdadeiros ficheiros de dados contínuos, quaisquer caracteres de formatação de linhas (tais como mudança de linha, tabulação e fim de ficheiro) são apenas convertidos de um CCSID para outro.

Quando está a ler a partir de ficheiros de registos que estão a ser utilizados como ficheiros de dados contínuos, os caracteres de fim de linha (mudança de linha) são acrescentados ao fim dos dados em cada registo. Quando o sistema escreve para ficheiros de registo:

- Os caracteres de fim de linha são removidos.
- Os caracteres de tabulação são substituídos pelo número apropriado de espaços em branco até à posição de tabulação seguinte.
- As linhas são preenchidas quer com espaços em branco (para um membro de ficheiro físico origem) quer com nulos (para um membro de ficheiro físico de dados) até ao fim do registo.

Num pedido de abertura, pode especificar uma das seguintes opções:

Binário, Forçado

Os dados são processados como binários, independentemente do conteúdo real do ficheiro. A aplicação é responsável pelo processo de tratamento dos dados.

Texto, Forçado

Assume-se que os dados são texto. Os dados são convertidos do CCSID do ficheiro para o CCSID da aplicação.

É utilizada a predefinição *Binário, Forçado* (*Binary, Forced*) para a função `open()` do sistema de ficheiros integrado.

Informações relacionadas

API `open()`

Exemplo: Funções C do sistema de ficheiros integrado

Este simples programa em linguagem C ilustra a utilização de várias funções do sistema de ficheiros integrado.

O programa executa as seguintes operações:

- 1 Utiliza a função `getuid()` para determinar o ID de utilizador (uid) real.
- 2 Utiliza a função `getcwd()` para determinar o directório actual.
- 3 Utiliza a função `open()` para criar um ficheiro. Concede a autoridade de leitura, escrita e execução do ficheiro ao proprietário (a pessoa que criou o ficheiro).
- 4 Utiliza a função `write()` para escrever uma cadeia de bytes no ficheiro. O descritor de ficheiro que foi fornecido na operação de abertura (3) identifica o ficheiro.
- 5 Utiliza a função `close()` para fechar o ficheiro.
- 6 Utiliza a função `mkdir()` para criar um novo subdirectório no directório actual. É concedido ao proprietário acesso de leitura, escrita e execução no subdirectório.
- 7 Utiliza a função `chdir()` para passar do novo subdirectório para o directório actual.

- 8 Utiliza a função link() para criar uma ligação ao ficheiro anteriormente criado (3).
- 9 Utiliza a função open() para abrir o ficheiro apenas para leitura. A ligação que foi criada em (8) permite o acesso ao ficheiro.
- 10 Utiliza a função read() para ler uma cadeia de bytes a partir do ficheiro. O descritor de ficheiro que foi fornecido na operação de abertura (9) identifica o ficheiro.
- 11 Utiliza a função close() para fechar o ficheiro.
- 12 Utiliza a função unlink() para remover a ligação ao ficheiro.
- 13 Utiliza a função chdir() para passar do directório actual novamente para o directório ascendente, onde o novo subdirectório foi criado.
- 14 Utiliza a função rmdir() para remover o subdirectório criado anteriormente (6).
- 15 Utiliza a função unlink() para remover o ficheiro criado anteriormente (3).

Nota: Este programa exemplo será correctamente executado nos sistemas em que o CCSID do trabalho que é executado seja 37. As APIs do sistema de ficheiros integrado têm de ter os nomes de objectos e de caminhos codificados no CCSID do trabalho; no entanto, o compilador de C armazena constantes de caracteres no CCSID 37. Para compatibilidade total, converta as constantes de caracteres, como por exemplo, nomes de objectos e caminhos, antes de transferir APIs para o CCSID do trabalho.

Nota: Ao utilizar os exemplos de código está implícita a aceitação dos termos do “Informações de licenças para códigos e de exclusão de responsabilidade” na página 149.

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

```
#define BUFFER_SIZE      2048
#define NEW_DIRECTORY    "testdir"
#define TEST_FILE        "test.file"
#define TEST_DATA        "Olá Mundo!"
#define USER_ID          "id_util_"
#define PARENT_DIRECTORY ".."
```

```
char InitialFile[BUFFER_SIZE];
char LinkName[BUFFER_SIZE];
char InitialDirectory[BUFFER_SIZE] = ".";
char Buffer[32];
int FilDes = -1;
int BytesRead;
int BytesWritten;
uid_t UserID;
```

```
void CleanupOnError(int level)
{
```

```
    printf("Foi encontrado um erro, a limpar.\n");
    switch ( level )
    {
        case 1:
            printf("Não foi possível determinar directório de trabalho actual.\n");
            break;
        case 2:
            printf("Não foi possível criar ficheiro %s.\n",TEST_FILE);
            break;
```

```

case 3:
    printf("Não foi possível gravar no ficheiro %s.\n",TEST_FILE);
    close(FilDes);
    unlink(TEST_FILE);
    break;
case 4:
    printf("Não foi possível fechar ficheiro %s.\n",TEST_FILE);
    close(FilDes);
    unlink(TEST_FILE);
    break;
case 5:
    printf("Não foi possível criar directório %s.\n",NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 6:
    printf("Não foi possível passar para o directório %s.\n", NEW_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 7:
    printf("Não foi possível criar ligação %s a %s.\n", LinkName,InitialFile);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 8:
    printf("Não foi possível abrir ligação %s.\n",LinkName);
    unlink(LinkName);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 9:
    printf("Não foi possível ler ligação %s.\n",LinkName);
    close(FilDes);
    unlink(LinkName);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 10:
    printf("Não foi possível fechar ligação %s.\n",LinkName);
    close(FilDes);
    unlink(LinkName);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 11:
    printf("Não foi possível remover ligação %s.\n",LinkName);
    unlink(LinkName);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 12:
    printf("Não foi possível passar para o directório %s.\n",
PARENT_DIRECTORY);
    chdir(PARENT_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;
case 13:
    printf("Não foi possível remover directório %s.\n",NEW_DIRECTORY);
    rmdir(NEW_DIRECTORY);
    unlink(TEST_FILE);
    break;

```

```

        case 14:
            printf("Não foi possível remover ficheiro %s.\n",TEST_FILE);
            unlink(TEST_FILE);
            break;
        default:
            break;
    }
    printf("O Programa terminou com Erros.\n"\
        "É possível que nem todos os ficheiros e directórios de teste tenham sido removidos.\n");
}

int main ()
{
    1
    /* Obtém e imprime o ID de utilizador real com a função getuid(). */
    UserID = getuid();
    printf("O verdadeiro ID de utilizador é %u.\n",UserID);

    2
    /* Obtém o directório de trabalho actual e armazena-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, gravar 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 acabado de criar. */
    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 a partir ficheiro LinkName, através de 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 através de FilDes. */
if ( -1 == close(FilDes) )
{
    perror("Erro na função close");
    CleanupOnError(10);
    return 0;
}
FilDes = -1;
printf("%s fechado.\n",LinkName);

12
/* Remove 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;
}

```

Referências relacionadas

“Executar operações utilizando APIs” na página 118

Muitas das interfaces de programas de aplicações (APIs) que executam operações em objectos do sistema de ficheiros integrado encontram-se sob a forma de funções de linguagem C.

Trabalhar com ficheiros e pastas através da utilização do iSeries Navigator

Pode executar estas tarefas com ficheiros e pastas.

Dar entrada de um ficheiro

Execute os passos que se seguem 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. Selecciona **Propriedades (Properties)**.
3. Selecciona **Propriedades do ficheiro (File Properties)** → **Utilizar página (Use Page)**.
4. Faça clique em **Dar entrada (Check In)**.

Dar saída de um ficheiro

Execute os passos seguintes para dar saída de um ficheiro.

1. No **iSeries Navigator**, faça clique com o botão direito do rato no ficheiro de que pretende dar saída.
2. Selecciona **Propriedades (Properties)**.
3. Faça clique em **Propriedades do ficheiro (File Properties)** → **Utilizar página (Use Page)**.
4. Faça clique em **Dar saída (Check Out)**.

Criar uma pasta

Execute os passos seguintes para criar uma pasta.

1. Expanda o sistema que pretende utilizar em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**.
2. Faça clique com o botão direito do rato sobre o sistema de ficheiros ao qual pretende adicionar a nova pasta e seccione **Nova pasta (New Folder)**.
3. Escreva um novo nome para o objecto na caixa de diálogo **Nova pasta (New Folder)**.
4. Faça clique em **OK**.

Quando cria uma pasta no servidor do iSeries, tem de considerar se pretende proteger a nova pasta (ou objecto) com a gestão de registos em diário. Também é necessário ter em consideração se pretende que os objectos criados nesta pasta sejam ou não pesquisados.

Tarefas relacionadas

“Definir se os objectos devem ou não ser pesquisados” na página 142

Execute os passos seguintes para definir se um objecto deve ou não ser pesquisado.

Informações relacionadas

Gestão de registos em diário

Remover uma pasta

Execute os passos seguintes para remover uma pasta.

1. Expanda o sistema que pretende utilizar em **iSeries Navigator** → **Sistema de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**. Continue a expandir até o ficheiro ou a pasta que pretende remover ficar visível.
2. Faça clique com o botão direito do rato no ficheiro ou na pasta e seccione **Eliminar (Delete)**.

Mover ficheiros ou pastas para outro sistema de ficheiros

Cada sistema de ficheiros tem as suas próprias características. No entanto, mover objectos para um sistema de ficheiros diferente pode significar a perda das vantagens do sistema de ficheiros em que os objectos estejam actualmente armazenados. Poderá pretender mover objectos de um sistema de ficheiros para outro para tirar partido 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.

Também deverá ter em conta a seguinte situação:

- 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 do sistema de ficheiros integrado. As aplicações que utilizam estas interfaces poderão já não aceder a objectos que foram movidos para outro sistema de ficheiros. Por exemplo, os sistemas de ficheiros QDLS e QOPT suportam as APIs do sistema hierárquico de ficheiros (HFS - Hierarchical File System) e os comandos para trabalhar com objectos de pastas e documentos. 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 obtenção apenas de alguns atributos expandidos, ao passo que os sistemas de ficheiros "raiz" (/) e QOpenSys suportam o armazenamento e obtenção de todos os atributos expandidos. Deste modo, QSYS.LIB e QSYS.LIB de ASP independente não são bons candidatos para o armazenamento de objectos com atributos expandidos.

Os ficheiros de PC armazenados no QDLS são bons candidatos para serem movidos. A maioria das aplicações de PC deverá poder continuar a trabalhar com os ficheiros de PC que forem movidos do QDLS para outros sistemas de ficheiros. Os sistemas de ficheiros "raiz" (/), QOpenSys, QNetWare e QNTC são boas opções para armazenar estes ficheiros de PC. Como suportam muitas das características do sistema de ficheiros OS/2, estes sistemas de ficheiros podem fornecer um acesso mais rápido a ficheiros.

Para mover objectos para outro sistema de ficheiros, execute os seguintes passos:

1. Guarde uma cópia de todos os objectos que tenciona mover.

A criação de uma cópia de segurança permite-lhe restaurar os objectos para o sistema de ficheiros original, caso verifique que as aplicações não conseguem aceder aos objectos no sistema de ficheiros para o qual foram movidos.

Nota: Não pode guardar objectos de um sistema de ficheiros e restaurá-los para outro.

2. Crie os directórios no sistema de ficheiros para o qual pretende mover os objectos, utilizando o comando Criar directório (CRTDIR).

Deve examinar com cuidado os atributos do directório onde os objectos se encontram actualmente para determinar se deseja duplicar esses atributos nos directórios que criar. Por exemplo, o utilizador que cria o directório é o respectivo proprietário, e não o utilizador proprietário do directório antigo. É conveniente transferir a propriedade do directório depois de o ter criado, se o sistema de ficheiros suportar a definição do proprietário de um directório.

3. Mova os ficheiros para o sistema de ficheiros seleccionado, utilizando o comando Mover (MOV).

Recomenda-se a utilização do comando MOV, uma vez que 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. Tenha em atenção que ao 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.

Conceitos relacionados

“Trabalhar com sistemas de ficheiros” na página 27

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.

Referências relacionadas

“Comparação entre sistemas de ficheiros” na página 29

Estas tabelas resumem as funções e limitações de cada sistema de ficheiros.

Informações relacionadas

Comando Criar directório (CRTDIR)

Comando Mover (MOV)

Comando Copiar (CPY)

Definir permissões

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

Para definir permissões para um ficheiro ou pasta, execute os seguintes passos:

1. Expanda o sistema que pretende utilizar em **iSeries Navigator** → **Sistema de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**. Continue a expandir até o objecto para o qual pretende adicionar permissões estar visível.
2. Faça clique com o botão direito do rato no objecto para o qual pretende adicionar permissões e seleccione **Permissões (Permissions)**.
3. Faça clique em **Adicionar (Add)** na caixa de diálogo **Permissões (Permissions)**.
4. Seleccione um ou mais utilizadores e grupos ou introduza o nome de um utilizador ou de um grupo no campo de nomes de utilizadores ou grupos, na caixa de diálogo **Adicionar (Add)**.
5. Faça clique em **OK**. Esta acção adiciona os utilizadores ou os grupos ao topo da lista.
6. Faça clique no botão **Detalhes (Details)** para implementar permissões detalhadas.
7. Aplique as permissões desejadas para o utilizador, seleccionando a caixa do quadrado de opção adequado.
8. Faça clique em **OK**.

Configurar conversão de texto do ficheiro

Pode configurar a conversão automática de ficheiros de texto no iSeries Navigator. A conversão automática de ficheiros de texto permite-lhe utilizar extensões de ficheiro para a conversão de dados de ficheiro.

O sistema de ficheiros integrado pode converter um ficheiro de dados quando é transferido entre um iSeries e um PC. Quando obtém acesso ao ficheiro de dados a partir de um PC, ele é tratado como se estivesse em ASCII.

Para configurar conversão de texto do ficheiro, execute estes passos:

1. Expanda o sistema que pretende utilizar em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)**.

2. Faça clique com o botão direito do rato sobre **Sistema de Ficheiros Integrado** e seleccione **Propriedades**.
3. 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**.
4. Repita o passo 3 para todas as extensões de ficheiro que pretenda converter automaticamente.
5. Faça clique em **OK**.

Enviar um ficheiro ou uma pasta para outro sistema

Execute os passos seguintes para enviar um ficheiro ou uma pasta para outro sistema.

1. Expanda o sistema que pretende utilizar em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**. Continue a expandir até o ficheiro ou a pasta que pretende enviar ficar visível.
2. Faça clique com o botão direito do rato no ficheiro ou na pasta e seleccione **Enviar (Send)**. O ficheiro ou a pasta é apresentado na lista **Ficheiros e pastas seleccionados (Selected Files and Folders)**, na caixa de diálogo **Enviar ficheiros a partir de (Send Files from)**.
3. Expanda a lista de sistemas e grupos disponíveis.
4. Seleccione um sistema e faça clique sobre **Adicionar**, para adicionar o sistema à lista **Sistemas e grupos destino**. Repita este passo para todos os sistemas para os quais pretenda enviar este ficheiro ou pasta.
5. Faça clique em **OK** para enviar o ficheiro ou a pasta com as actuais definições do pacote e informações da marcação predefinidas.

Quando cria uma definição de pacote, ela é guardada e pode ser reutilizada em qualquer momento para enviar o conjunto de ficheiros e pastas definidos para vários sistemas ou grupos de sistemas de ponto de terminação. Se optar por criar um instantâneo dos seus ficheiros, pode manter mais do que uma versão das cópias do mesmo conjunto de ficheiros. O envio do instantâneo assegura que não serão efectuadas alterações aos ficheiros durante a distribuição, de modo a que o último sistema destino receba os mesmos objectos que o primeiro sistema destino.

Tarefas relacionadas

“Alterar opções para a definição do pacote”

Uma definição de pacote permite agrupar um conjunto de objectos ou ficheiros do sistema de ficheiros integrado do i5/OS.

“Marcar uma data e uma hora para enviar o ficheiro ou a pasta” na página 140

A função de agendamento oferece ao utilizador flexibilidade para executar o respectivo trabalho quando lhe for conveniente.

Alterar opções para a definição do pacote

Uma definição de pacote permite agrupar um conjunto de objectos ou ficheiros do sistema de ficheiros integrado do i5/OS.

A definição de pacote também lhe permite visualizar este mesmo grupo de ficheiros como um conjunto lógico, ou como um conjunto físico, tirando um instantâneo dos ficheiros de modo a preservá-los para distribuição posterior.

Para alterar as opções das definições de pacote, siga os seguintes passos:

1. Execute os passos incluídos em “Enviar um ficheiro ou uma pasta para outro sistema”.
2. Faça clique no separador **Opções (Options)**. As opções predefinidas são incluir subpastas ao condensar e enviar ficheiros e substituir um ficheiro existente pelo enviado.
3. Altere estas opções se necessário.
4. Faça clique em **Avançada (Advanced)** para definir as opções de salvaguarda e restauro avançadas.
5. Faça clique em **OK** para guardar as opções avançadas.

6. Faça clique em **OK** para enviar o ficheiro ou **Agendar (Schedule)**, para definir uma hora para enviar o ficheiro.

Tarefas relacionadas

“Enviar um ficheiro ou uma pasta para outro sistema” na página 139

Execute os passos seguintes para enviar um ficheiro ou uma pasta para outro sistema.

“Marcar uma data e uma hora para enviar o ficheiro ou a pasta”

A função de agendamento oferece ao utilizador flexibilidade para executar o respectivo trabalho quando lhe for conveniente.

Marcar uma data e uma hora para enviar o ficheiro ou a pasta

A função de agendamento oferece ao utilizador flexibilidade para executar o respectivo trabalho quando lhe for conveniente.

Para marcar uma data e uma hora para enviar o ficheiro ou a pasta:

1. Execute os passos incluídos em “Enviar um ficheiro ou uma pasta para outro sistema” na página 139.
2. Faça clique sem **Agendar(Schedule)**.
3. Selecciona as opções relativas à altura em que pretende enviar o ficheiro ou a pasta.

Tarefas relacionadas

“Enviar um ficheiro ou uma pasta para outro sistema” na página 139

Execute os passos seguintes para enviar um ficheiro ou uma pasta para outro sistema.

“Alterar opções para a definição do pacote” na página 139

Uma definição de pacote permite agrupar um conjunto de objectos ou ficheiros do sistema de ficheiros integrado do i5/OS.

Criar uma partilha de ficheiros

Uma *partilha de ficheiros* consiste num caminho do directório que o iSeries NetServer partilha com clientes de PC na rede do iSeries. Uma partilha de ficheiros pode ser constituída por qualquer directório do sistema de ficheiros integrado do iSeries.

Para criar uma partilha de ficheiros, execute os passos seguintes.

1. Expanda o sistema em **iSeries Navigator** → **Sistema de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**.
2. Expanda o sistema de ficheiros que contém a pasta para o qual pretende criar uma partilha.
3. Faça clique com o botão direito do rato na pasta para a qual pretende criar uma partilha e seccione **Partilhar (Share)**.
4. Seccione **Nova partilha (New Share)**.

Alterar uma partilha de ficheiros

Uma *partilha de ficheiros* consiste num caminho do directório que o iSeries NetServer partilha com clientes de PC na rede do iSeries. Uma partilha de ficheiros pode ser constituída por qualquer directório do sistema de ficheiros integrado do iSeries.

Para alterar uma partilha de ficheiros, execute os passos seguintes.

1. Expanda o sistema em **iSeries Navigator** → **Sistema de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**.
2. Expanda a pasta com a partilha definida que pretende alterar.
3. Faça clique com o botão direito do rato na pasta com a partilha definida que pretende **Partilhar (Share)**.
4. Seccione **Nova partilha (New Share)**.

Criar um novo sistema de ficheiros definido pelo utilizador

Um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System) é um sistema de ficheiros para o qual são criados e definidos os atributos. Os UDFSs residem em áreas de memória auxiliar (ASP - Auxiliary Storage Pools) no sistema.

Para criar um novo sistema de ficheiros definido pelo utilizador (UDFS), execute os seguintes passos:

1. Expanda o sistema em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)** → **Raiz (Root)** → **Dev.**
2. Faça clique na área de memória auxiliar (ASP) que pretende que contenha os novos UDFS.
3. Selecciona **Novo UDFS (New UDFS)** a partir do menu Ficheiro (File).
4. Especifique o nome de UDFS, a descrição (opcional), os valores de auditoria, o formato de ficheiro predefinido, o atributo de pesquisa predefinido e se os ficheiros no novo UDFS terão nomes de ficheiros sensíveis a maiúsculas e minúsculas na caixa de diálogo Novo sistema de ficheiros definido pelo utilizador (New User-Defined File System).

Montar sistema de ficheiros definido pelo utilizador

Para aceder ou visualizar os dados armazenados num sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), tem de montar o UDFS após cada IPL.

Ao montar um UDFS, são abrangidos todos os sistemas de ficheiros, directórios ou objectos que existam sob o ponto de montagem na hierarquia de pastas. Este processo torna esses sistemas de ficheiros, directórios ou objectos inacessíveis até o UDFS ser desmontado. Para se certificar de que o acesso a todos os dados do sistema de ficheiros integrado é mantido, monte o UDFS acima de uma pasta vazia. Após um UDFS ser montado, os ficheiros no UDFS estarão acessíveis a partir dessa pasta. Quaisquer alterações efectuadas na pasta constituirão alterações ao UDFS e não à pasta abrangida.

Nota: Não pode montar um UDFS por cima de outro numa ASP independente.

Para montar um sistema de ficheiros definido pelo utilizador (UDFS), execute os seguintes passos:

1. Expanda o sistema em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)** → **Raiz (Root)** → **Dev.**
2. Faça clique na área de memória auxiliar (ASP) que contém o UDFS que pretende montar.
3. Faça clique com o botão direito do rato no UDFS que pretende montar na coluna **Nome do UDFS (UDFS Name)** da área da janela direita do Navegador de operações (Operations Navigator).
4. Selecciona **Montar (Mount)**.

Caso pretenda utilizar a função arrastar e largar, pode montar um UDFS arrastando-o para uma pasta do sistema de ficheiros integrado no mesmo servidor. Não pode largar o UDFS em /dev, /dev/QASPxx, /dev/nome_asp, noutra sistema, nem na área de trabalho.

Desmontar sistema de ficheiros definido pelo utilizador

Ao montar um UDFS, são abrangidos todos os sistemas de ficheiros, directórios ou objectos que existam sob o ponto de montagem na hierarquia de pastas. Este processo torna esses sistemas de ficheiros, directórios ou objectos inacessíveis até o UDFS ser desmontado.

Para desmontar um sistema de ficheiros definido pelo utilizador (UDFS - User-Defined File System), execute os seguintes passos:

1. Expanda o sistema em **iSeries Navigator** → **Sistemas de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)** → **Raiz (Root)** → **Dev.**
2. Faça clique no conjunto de memória auxiliar (ASP) que contém o UDFS que pretende desmontar.
3. Faça clique com o botão direito do rato no UDFS que pretende desmontar na coluna **Nome do UDFS (UDFS Name)** da área da janela direita do iSeries Navigator.

4. Selecione **Desmontar (Unmount)**.

Definir se os objectos devem ou não ser pesquisados

Execute os passos seguintes para definir se um objecto deve ou não ser pesquisado.

1. Expanda o sistema em **iSeries Navigator** → **Sistema de ficheiros (File Systems)** → **Sistema de ficheiros integrado (Integrated File System)**.
2. Expanda a pasta ou o ficheiro pretendido.
3. Faça clique com o botão direito do rato na pasta ou o ficheiro e selecione **Propriedades (Properties)**
4. Selecione o separador **Segurança (Security)**.
5. Selecione **Pesquisar objectos (Scan Objects)** com a opção pretendida.

Para obter mais informações sobre as opções, consulte as secções que se seguem. As descrições destas opções dizem respeito aos ficheiros. Só os ficheiros podem ser pesquisados. Nas pastas e sistemas de ficheiros definidos pelo utilizador, pode especificar o atributo de pesquisa a ser atribuído a ficheiros criados nessa pasta ou no sistema de ficheiros definido pelo utilizador.

- Sim

O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado ou se o software de pesquisa tiver sido actualizado desde a última pesquisa efectuada ao objecto.

- Não

O objecto não será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa.

Nota: Se a opção Pesquisar no próximo acesso após restauro do objecto (Scan on next access after object has been restored) estiver seleccionada nos valores de sistema, quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.

- Só quando o objecto foi alterado

O objecto será pesquisado de acordo com as regras descritas nos programas de saída relacionados com a pesquisa se o objecto tiver sido modificado desde a última pesquisa efectuada ao objecto. Não será efectuada a pesquisa se o software de pesquisa tiver sido actualizado.

Se a opção Utilizar apenas quando os objectos tiverem alterado o atributo para pesquisa de controlo (Use only when objects have changed attribute to control scan) não estiver especificada nos valores de sistema, não será utilizado este atributo alteração apenas de objecto e o objecto será pesquisado após ter sido modificado e quando o software de pesquisa indicar uma actualização.

Notas:

1. Neste separador de ficheiros, pode também determinar o estado de pesquisa de um objecto.
2. Se a opção Pesquisar no próximo acesso após restauro do objecto (Scan on next access after object has been restored) estiver seleccionada nos valores de sistema, quando um objecto com este atributo for restaurado, o objecto será pesquisado pelo menos uma vez após o restauro.

Conceitos relacionados

“Suporte de pesquisa” na página 21

Com o iSeries, pode pesquisar objectos do sistema de ficheiros integrado.

Tarefas relacionadas

“Criar uma pasta” na página 136

Execute os passos seguintes para criar uma pasta.

Chamada de procedimento remoto independente de transporte

Desenvolvida pela Sun Microsystems, a chamada de procedimento remoto (RPC - Remote Procedure Call) separa e distribui facilmente aplicações de cliente a partir de um mecanismo de servidor.

A RPC inclui uma norma para a representação de dados, denominada eXternal Data Representation (XDR), de modo a permitir que mais de um tipo de máquina acesse dados transmitidos. A RPC independente de transporte (TI-RPC) é a versão mais recente de RPC. Fornece um método para separar o protocolo subjacente que é utilizado ao nível da rede, possibilitando uma transição mais uniforme de um protocolo para outro. Os únicos protocolos que presentemente estão disponíveis no servidor são TCP e UDP.

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

APIs de selecção de rede

Estas APIs fornecem o meio de selecção do transporte, no qual 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
<code>endnetconfig()</code>	Liberta o indicador para os registos armazenados no ficheiro <code>netconfig</code>
<code>freenetconfignt()</code>	Liberta a estrutura do <code>netconfig</code> que é devolvida da chamada à função <code>getnetconfignt()</code>
<code>getnetconfig()</code>	Devolve o indicador ao registo actual no ficheiro <code>netconfig</code> e incrementa o respectivo indicador para o registo seguinte
<code>getnetconfignt()</code>	Devolve o indicador à estrutura do <code>netconfig</code> que corresponde ao <code>netid</code> de entrada
<code>setnetconfig()</code>	Inicializa o indicador de registo para a primeira entrada do ficheiro <code>netconfig</code> . A função <code>setnetconfig()</code> tem de ser utilizada antes da função <code>getnetconfig()</code> ser utilizada pela primeira vez. A função <code>setnetconfig()</code> devolve um único parâmetro identificador (um apontador para os registos armazenados no ficheiro <code>netconfig</code>) a ser utilizado pela função <code>getnetconfig()</code> .

Informações relacionadas

Localizador de APIs

APIs de conversão de nomes para endereços

Estas APIs permitem que uma aplicação obtenha o endereço de um serviço ou de um sistema central especificado de uma forma independente de transporte.

API	Descrição
<code>netdir_free()</code>	Liberta estruturas atribuídas por APIs de conversão de nome para endereço
<code>netdir_getbyaddr()</code>	Define endereços para nomes de sistemas centrais e de serviços
<code>netdir_getbyname()</code>	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 <code>netconfig</code>

API	Descrição
netdir_options()	Fornecer interfaces para capacidades específicas do transporte como, por exemplo, o endereço de difusão e funções de porta reservada de TCP e UDP
netdir_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)

Informações relacionadas

Localizador de APIs

APIs XDR (eXternal Data Representation)

Estas APIs permitem que aplicações RPC processem estruturas de dados arbitrárias, independentemente das diferentes ordens de bytes do sistema central ou de convenções de esquema de estruturas.

API	Descrição
xdr_array()	Um valor primitivo de filtro que faz a conversão entre matrizes de comprimento variável e as respectivas representações externas correspondentes. Esta função é chamada para codificar ou decodificar cada elemento da matriz
xdr_bool()	Um valor primitivo de filtro que faz a conversão entre valores Booleanos (números inteiros em C) e as respectivas representações externas. Ao codificar dados, este filtro produz valores iguais a 1 ou 0.
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

API	Descrição
xdr_opaque()	Um valor primitivo de filtro que faz a conversão entre dados opacos de tamanho fixo e a respectiva representação externa
xdr_pointer()	Possibilita a procura de indicadores em estruturas e coloca em série indicadores nulos. Pode representar estruturas de dados recursivas como, por exemplo, árvores binárias ou listas ligadas.
xdr_reference()	Um valor primitivo de filtro que possibilita a procura de indicadores em estruturas. Este valor primitivo permite colocar em série, retirar de série e libertar quaisquer indicadores, numa estrutura, que sejam referidos por outra estrutura. A função xdr_reference() não associa qualquer significado especial a um indicador nulo durante a serialização e ao transferir o endereço de um indicador nulo poderá causar um erro de memória. Por este motivo, o programador tem de descrever os dados com uma união bilateral discriminada. Um lado é utilizado quando o indicador é válido; o outro lado, quando o indicador é nulo.
xdr_short()	Um valor primitivo de filtro que faz a conversão entre números inteiros curtos de linguagem C e a respectiva representação externa
xdr_string()	Um valor primitivo de filtro que faz a conversão entre cadeias de linguagem C e as respectivas representações externas
xdr_u_char()	Um valor primitivo de filtro que faz a conversão entre caracteres de linguagem C sem sinal e as respectivas representações externas
xdr_u_int()	Um valor primitivo de filtro que faz a conversão entre números inteiros sem sinal de linguagem C e as respectivas representações externas
xdr_u_long()	Um valor primitivo de filtro que faz a conversão entre números inteiros longos sem sinal de linguagem C e as respectivas representações externas
xdr_u_short()	Um valor primitivo de filtro que faz a conversão entre números inteiros curtos sem sinal de linguagem C e as respectivas representações externas
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), em que maxuint corresponde ao valor máximo de um número inteiro sem sinal. xdr_wrapstring() é útil, uma vez que o pacote de RPC transfere um máximo de duas funções de XDR como parâmetros e a função xdr_string() requer três.

Informações relacionadas

Localizador de APIs

APIs de autenticação

Estas APIs fornecem autenticação às aplicações TI-RPC.

API	Descrição
auth_destroy()	Destrói a estrutura das informações de autenticação que é indicada pelo parâmetro de autenticação
authnone_create()	Cria e devolve um parâmetro identificador de autenticação de RPC predefinido que transfere informações de autenticação nulas em cada chamada de procedimento remoto.

API	Descrição
authsys_create()	Cria e devolve um parâmetro identificador de autenticação de RPC que contém informações de autenticação

Informações relacionadas

Localizador de APIs

APIs de RPC independente de transporte (TI-RPC)

Estas APIs fornecem um ambiente de programação de aplicações distribuídas isolando a aplicação face a qualquer componente de transporte específico. Esta função facilita a utilização de transportes.

Informações relacionadas

Localizador de APIs

APIs simplificadas TI-RPC

Estas APIs simplificadas 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

Informações relacionadas

Localizador de APIs

APIs de nível superior TI-RPC

Estas APIs 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

Informações relacionadas

Localizador de APIs

APIs de nível intermédio TI-RPC

Estas APIs são semelhantes às APIs de nível superior, mas as aplicações do utilizador seleccionam as informações específicas de 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

Informações relacionadas

Localizador de APIs

APIs ao nível de perito TI-RPC

Estas APIs permitem que a aplicação seleccione o transporte a utilizar. Também fornecem um nível de controlo melhorado sobre os detalhes dos parâmetros identificadores CLIENT e SVCXPRT. Estas APIs são semelhantes às APIs de nível intermédio com um controlo adicional fornecido, utilizando as APIs de conversão de nome para endereço.

É fornecido um controlo adicional, utilizando as APIs de conversão de nome para endereço.

API	Descrição
clnt_tli_create()	Cria um parâmetro identificador de cliente
rpcb_getaddr()	Localiza o endereço universal de um serviço
rpcb_set()	Regista o endereço do servidor com RPCbind
rpcb_unset()	É utilizada pelos servidores para anular do registo os respectivos endereços
svc_reg()	Associa o programa e a versão ao envio
svc_tli_create()	Cria um parâmetro identificador de servidor
svc_unreg()	Elimina um conjunto de associações por svc_reg()

Informações relacionadas

Localizador de APIs

Outras APIs TI-RPC

Estas APIs permitem que várias aplicações funcionem em coordenação com as APIs simplificadas, de nível superior, de nível intermédio e de nível de perito.

API	Descrição
clnt_freeres()	Liberta os dados atribuídos pela RPC ou pelo sistema XDR
clnt_geterr()	Obtém a estrutura de erro a partir do parâmetro identificador do cliente
svc_freeargs()	Liberta os dados atribuídos pela RPC ou pelo sistema XDR
svc_getargs()	Descodifica argumentos de um pedido de RPC
svc_getrpcaller()	Obtém o endereço de rede do programa de chamada
svc_run()	Aguarda a entrada de pedidos de RPC
svc_sendreply()	Envia os resultados de uma chamada de procedimento remoto para um cliente remoto.
svcerr_decode()	Envia informações para o cliente sobre o erro de descodificação
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

Informações relacionadas

Localizador de APIs

Informações relacionadas com o sistema de ficheiros integrado

Nesta secção é possível encontrar manuais de produtos, sítios da Web e tópicos do centro de informações relacionados com o tópico sobre o sistema de ficheiros integrado. Pode visualizar ou imprimir qualquer dos PDFs.

Manuais

- Suporte de Network File System para o i5/OS  Este manual descreve o Network File System através de uma série de aplicações reais. 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.

- Suporte Óptico  Este manual funciona como orientação e referência do utilizador no que respeita ao suporte óptico da IBM no i5/OS. As informações deste manual podem ajudar o utilizador a compreender os conceitos do servidor de dados da biblioteca óptica, a planear uma biblioteca óptica, a administrar e a funcionar com um servidor de dados da biblioteca óptica e a resolver problemas no servidor de dados ópticos.
- WebSphere Development Studio: Referência à linguagem C/C++  Este manual fornece informações necessárias para conceber, editar, compilar, executar e depurar programas de ILE C/400 no servidor iSeries.
- Segurança — Referência  Este manual fornece informações técnicas detalhadas sobre segurança do i5/OS incluindo valores de sistema relacionados com a segurança com impacto no processamento relacionado com a pesquisa do sistema de ficheiros integrado.
- Programação APPC  Este manual descreve o suporte APPC (Advanced Program-to-Program Communications) 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 Restauro  Este manual fornece informações gerais sobre as opções de recuperação e disponibilidade relativas ao servidor IBM iSeries.

Outras informações

• Relatórios de experiências

Os relatórios de experiências são escritos por programadores da IBM, que documentam as suas experiências na resolução de problemas, implementando cenários e soluções reais. Utilize os relatórios para acompanhar as experiências dos programadores da IBM na implementação específica de uma solução do iSeries, incluindo instruções passo a passo e sugestões. O relatório de experiência Efectuar cópia de segurança do sistema de ficheiros integrado está relacionado com ficheiros e sistemas de ficheiros.

- Control language
- Globalização do i5/OS
- Interfaces de programação de aplicações (APIs)
- Gestão de registos em diário
- Controlo de consolidações

Guardar ficheiros PDF

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

1. No navegador, faça clique com o botão direito do rato no PDF pretendido (faça clique com o botão direito do rato na ligação acima).
2. Faça clique na opção que permite guardar o PDF localmente.
3. Navegue para o directório no qual pretende guardar o PDF.
4. Faça clique em **Save** (Guardar).

Descarregar o Adobe Reader

- | Necessita de ter o Adobe Reader instalado no sistema para visualizar ou imprimir estes PDFs. Pode
- | descarregar uma cópia gratuita a partir do sítio da Web da Adobe
- | (www.adobe.com/products/acrobat/readstep.html) .

Informações de licenças para códigos e de exclusão de responsabilidade

A IBM concede-lhe uma licença de direitos de autor não exclusiva para a utilização de todos os exemplos de códigos de programação, a partir dos quais é possível gerar funções semelhantes adequadas às suas necessidades específicas.

| SUJEITOS A QUAISQUER GARANTIAS ESTATUTÁRIAS QUE NÃO POSSAM SER EXCLUÍDAS, A IBM,
| OS SEUS PROGRAMADORES E FORNECEDORES DE PROGRAMAS NÃO FORNECEM GARANTIAS
| OU CONDIÇÕES DE QUALQUER ESPÉCIE, EXPLÍCITAS OU IMPLÍCITAS, INCLUINDO MAS NÃO SE
| LIMITANDO ÀS CONDIÇÕES OU GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO
| A UM DETERMINADO FIM E NÃO INFRACÇÃO, RELATIVAMENTE AO PROGRAMA E AO
| SUPORTE TÉCNICO, SE EXISTIR.

| EM NENHUMA CIRCUNSTÂNCIA A IBM, OS SEUS PROGRAMADORES OU FORNECEDORES DE
| PROGRAMAS SÃO RESPONSÁVEIS PELO SEGUINTE, AINDA QUE INFORMADOS DA SUA
| POSSIBILIDADE:

- | 1. PERDA OU DANOS DE DADOS;
- | 2. DANOS DIRECTOS, ESPECIAIS, INCIDENTAIS OU INDIRECTOS, OU QUALQUER DANO
| ECONÓMICO CONSEQUENCIAL; OU
- | 3. LUCROS CESSANTES, PERDA DE NEGÓCIOS, RECEITAS, CLIENTELA OU POUPANÇAS
| ANTECIPADAS.

| ALGUMAS JURISDIÇÕES NÃO PERMITEM A EXCLUSÃO OU LIMITAÇÃO DOS DANOS DIRECTOS,
| INCIDENTAIS OU CONSEQUENCIAIS, PELO QUE ALGUMAS OU TODAS ESTAS EXCLUSÕES OU
| LIMITAÇÕES PODERÃO NÃO SER APLICÁVEIS AO SEU CASO.

Apêndice. Avisos

Os produtos, serviços ou componentes descritos neste documento poderão não ser disponibilizados pela IBM noutros países. Consulte o representante da IBM para obter informações sobre os produtos e serviços actualmente disponíveis na sua área. Quaisquer referências, nesta publicação, a produtos, programas ou serviços da IBM, não significam que apenas esses produtos, programas ou serviços da IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. No entanto, é da inteira responsabilidade do utilizador avaliar e verificar o funcionamento de qualquer produto, programa ou serviço não produzido pela IBM.

Neste documento, podem ser feitas referências a patentes ou a pedidos de patente pendentes da IBM. O facto de este documento lhe ser fornecido não lhe confere nenhum direito sobre essas patentes. Caso solicite pedidos de informação sobre licenças, tais pedidos deverão ser endereçados, por escrito, para:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
E.U.A.

O parágrafo seguinte não se aplica ao Reino Unido nem a nenhum outro país onde estas cláusulas sejam inconsistentes com a lei local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "TAL COMO ESTÁ" (AS IS), SEM GARANTIA DE QUALQUER ESPÉCIE, EXPLÍCITA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE NÃO INFRACÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO FIM. Alguns Estados não permitem a exclusão de garantias, explícitas ou implícitas, em determinadas transacções; esta declaração pode, portanto, não se aplicar ao seu caso.

Esta publicação pode conter imprecisões técnicas ou erros de tipografia. A IBM permite-se fazer alterações periódicas às informações aqui contidas; essas alterações serão incluídas nas posteriores edições desta publicação. A IBM pode introduzir melhorias e/ou alterações ao(s) produto(s) e/ou programa(s) descrito(s) nesta publicação em qualquer momento, sem aviso prévio.

Quaisquer referências, nesta publicação, a sítios da Web que não sejam propriedade da IBM são fornecidas apenas para conveniência e não constituem, em caso algum, aprovação desses sítios da Web. Os materiais existentes nesses sítios da Web não fazem parte dos materiais destinados a este produto IBM e a utilização desses sítios da Web será da exclusiva responsabilidade do utilizador.

A IBM pode usar ou distribuir quaisquer informações que lhe forneça, da forma que julgue apropriada, sem incorrer em nenhuma obrigação para consigo.

Os licenciados deste programa que pretendam obter informações sobre o mesmo com o objectivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização recíproca das informações que tenham sido trocadas, deverão contactar:

Companhia IBM Portuguesa, S.A.
Edifício Office Oriente
Rua do Mar da China, Lote 1.07.2.3
Parque das Nações
1990-039 Lisboa

Tais informações poderão estar disponíveis, sujeitas aos termos e às condições adequadas, incluindo, em alguns casos, o pagamento de um encargo.

O programa licenciado descrito nestas informações e todo o material licenciado disponível para o programa são fornecidos pela IBM nos termos das Condições Gerais IBM (IBM Customer Agreement), Acordo de Licença Internacional para Programas IBM (IPLA, IBM International Program License Agreement), Acordo de Licença para Código Máquina IBM (IBM License Agreement for Machine Code) ou de qualquer acordo equivalente entre ambas as partes.

Quaisquer dados de desempenho aqui contidos foram determinados num ambiente controlado. Assim sendo, os resultados obtidos noutros ambientes operativos podem variar significativamente. Algumas medições podem ter sido efectuadas em sistemas ao nível do desenvolvimento, pelo que não existem garantias de que estas medições sejam iguais nos sistemas disponíveis habitualmente. Para além disso, algumas medições podem ter sido calculadas por extrapolação. Os resultados reais podem variar. Os utilizadores deste documento devem verificar os dados aplicáveis ao seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto dos fornecedores desses produtos, dos seus anúncios publicados ou de outras fontes de divulgação ao público. A IBM não testou esses produtos e não pode confirmar a exactidão do desempenho, da compatibilidade ou de quaisquer outras afirmações relacionadas com produtos não IBM. Todas as questões sobre as capacidades dos produtos não produzidos pela IBM deverão ser endereçadas aos fornecedores desses produtos.

Todas as afirmações relativas às directivas ou tendências futuras da IBM estão sujeitas a alterações ou descontinuação sem aviso prévio, representando apenas metas e objectivos.

Estas informações contêm exemplos de dados e relatórios utilizados em operações comerciais diárias. Para ilustrá-los o melhor possível, os exemplos incluem nomes de indivíduos, firmas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e moradas reais é mera coincidência.

LICENÇA DE COPYRIGHT:

Esta publicação contém programas de aplicação exemplo em linguagem-fonte, que ilustram técnicas de programação em diversas plataformas operativas. Pode copiar, modificar e distribuir estes programas exemplo de qualquer forma, sem encargos para com a IBM, com a finalidade de desenvolver, utilizar, comercializar ou distribuir programas de aplicação conformes à interface de programação de aplicações e destinados à plataforma operativa para a qual os programas exemplo são escritos. Estes exemplos não foram testados exaustivamente sob todas as condições. Deste modo, a IBM não garante nem se responsabiliza pela fiabilidade, assistência ou funcionamento destes programas.

Cada cópia ou qualquer parte destes programas exemplo ou qualquer trabalho derivado dos mesmos tem de incluir um aviso de direitos de autor, do seguinte modo:

© (o nome da sua empresa) (ano). Algumas partes deste código são derivadas de Programas Exemplo da IBM Corp. © Copyright IBM Corp. _introduza o(s) ano(s). Todos os direitos reservados.

Se estiver a consultar a versão electrónica desta publicação, é possível que as fotografias e as ilustrações a cores não estejam visíveis.

Informações sobre Interfaces de Programação

Esta publicação sobre o sistema de ficheiros integrado documenta futuras interfaces de programação que permitem ao cliente gravar programas para obter os serviços do IBM i5/OS.

Marcas Comerciais

Os termos seguintes são marcas comerciais da International Business Machines Corporation nos Estados Unidos e/ou noutros países:

C/400
DB2
i5/OS
IBM
IBM (logotipo)
Integrated Language Environment
iSeries
NetServer
OfficeVision
OS/2
OS/400
WebSphere
xSeries

Microsoft, Windows, Windows NT e o logotipo do Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou noutros países.

Linux é uma marca comercial da Linus Torvalds nos Estados Unidos e/ou noutros países.

UNIX é uma marca comercial registada de The Open Group nos Estados Unidos e noutros países.

Outros nomes de empresas, produtos e serviços podem ser marcas comerciais ou marcas de serviço de terceiros.

Termos e Condições

As permissões de utilização destas publicações são concedidas sujeitas aos termos e condições seguintes.

Utilização pessoal: Pode reproduzir estas publicações para uso pessoal e não comercial, desde que mantenha todas as informações de propriedade. Não pode executar qualquer trabalho derivado destas publicações, nem reproduzir, distribuir ou apresentar qualquer parte das mesmas, sem o expresse consentimento da IBM.

Utilização comercial: Pode reproduzir, distribuir e apresentar estas publicações exclusivamente no âmbito da sua empresa, desde que mantenha todas as informações de propriedade. Não pode executar qualquer trabalho derivado destas publicações, nem reproduzir, distribuir ou apresentar estas publicações, ou qualquer parte das mesmas fora das instalações da empresa, sem o expresse consentimento da IBM.

À excepção das concessões expressas nesta permissão, não são concedidos outros direitos, permissões ou licenças, quer explícitos, quer implícitos, sobre as publicações ou quaisquer informações, dados, software ou outra propriedade intelectual contidos nesta publicação.

A IBM reserva-se o direito de retirar as permissões concedidas nesta publicação sempre que considerar que a utilização das publicações pode ser prejudicial aos seus interesses ou, tal como determinado pela IBM, sempre que as instruções acima referidas não estejam a ser devidamente cumpridas.

Não pode descarregar, exportar ou reexportar estas informações, excepto quando em total conformidade com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação em vigor nos Estados Unidos.

A IBM NÃO FORNECE QUAISQUER GARANTIAS RELATIVAMENTE AO CONTEÚDO DESTAS PUBLICAÇÕES. AS PUBLICAÇÕES SÃO FORNECIDAS "TAL COMO ESTÃO" E SEM GARANTIAS DE QUALQUER ESPÉCIE, QUER EXPLÍCITAS, QUER IMPLÍCITAS, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, NÃO INFRACÇÃO E ADEQUAÇÃO A UM DETERMINADO FIM.

IBM