IBM Spectrum Protect
Versão 8.1.0

Usando a Interface de Programação de Aplicativos

IBM Spectrum Protect
Versão 8.1.0

Usando a Interface de Programação de Aplicativos

Nota:
Antes de usar estas informações e o produto que elas suportam, leia as informações em “Aviso” na página 219.

Esta edição se aplica à versão 8, liberação 1, modificação 0 de IBM Spectrum Protect (números de produto 5725-W98, 5725-W99, e 5725-X15) e para todas as liberações e modificações subsequentes até que indicado de outra forma em novas edições.

Índice

Sobre Esta Publicação

Esta publicação fornece informações para ajudá-lo a desempenhar as seguintes tarefas:

- Incluir chamadas de interface de programa de aplicativo do IBM Spectrum Protect em um aplicativo existente
- Gravar programas com interfaces de programa de uso geral que obtenham os serviços do IBM Spectrum Protect.

Além da API (Interface de Programação de Aplicativos), os seguintes programas estão incluídos nos vários sistemas operacionais:

- Um programa cliente de backup/archive que faz backup e arquiva arquivos da estação de trabalho ou do servidor de arquivo para armazenamento e restaura e recupera versões de backup e cópias arquivadas de arquivos para sistemas de arquivos locais.
- Um cliente de backup/archive da Web que um administrador, um atendente de suporte técnico ou um usuário final autorizado pode utilizar para executar serviços de backup, restauração, archive e recuperação utilizando um navegador da Web em uma máquina remota.
- Um programa cliente administrativo que você pode acessar em um navegador da Web ou na linha de comandos. Um administrador controla e monitora atividades do servidor, define políticas de gerenciamento de armazenamento para serviços de backup, archive e gerenciamento de espaço e configura planejamentos para executar esses serviços em intervalos regulares.

Quem Deve Ler Esta Publicação

Esta publicação fornece instruções para incluir chamadas de API em um aplicativo existente. Você deve estar familiarizado com a linguagem de programação C e as funções do IBM Spectrum Protect.

Publicações

A família de produtos IBM Spectrum Protect inclui o IBM Spectrum Protect Snapshot, IBM Spectrum Protect for Space Management, IBM Spectrum Protect for Databases e vários outros produtos de gerenciamento de armazenamento da IBM®.

Para visualizar a documentação do produto IBM, consulte IBM Knowledge Center.

Convenções Usadas Nesta Publicação

Essa publicação usa as seguintes convenções de nomenclaturas tipográficas:

Exemplo	Descrição
autoexec.ncf hsmgui.exe	Uma série de letras minúsculas com uma extensão indica nomes de arquivos de programas.
DSMI_DIR	Uma série de letras maiúsculas indica códigos de retorno e outros valores.
dsmQuerySessInfo	O tipo negrito indica um comando que você digita em uma linha de comandos, o nome de uma chamada de função, o nome de uma estrutura, um campo dentro de uma estrutura ou um parâmetro.

Exemplo	Descrição
timeformat	Tipo em negrito e itálico indica uma opção do cliente de backup-archive. O tipo em negrito é usado para apresentar a opção ou em um exemplo.
dateformat	O tipo itálico indica uma opção, o valor de uma opção, um novo termo, um marcador para informações fornecidas por você ou para dar ênfase especial ao texto.
`maxcmdretries`	O tipo monoespaçamento indica fragmentos de um programa ou informações à medida que são exibidos em uma tela, como um exemplo de comando.
sinal de mais (+)	Um sinal de mais entre duas teclas indica que as duas teclas devem ser pressionadas ao mesmo tempo.

O que há de novo nas da Versão 8.1.0

IBM Spectrum Protect Versão 8.1.0 introduz novos recursos e atualizações.

Para obter uma lista de novos recursos e atualizações nesta liberação, consulte Atualizações de API.

Capítulo 1. Visão Geral da API

A API (Interface de Programação de Aplicativo) do IBM Spectrum Protect permite que um cliente aplicativo utilize as funções de gerenciamento de armazenamento.

A API inclui chamadas de funções que você pode utilizar em um aplicativo para desempenhar as seguintes operações:
- Iniciar ou finalizar uma sessão
- Designar classes de gerenciamento a objetos antes que eles sejam armazenados em um servidor
- Fazer backup ou archive de objetos para um servidor
- Restaurar ou recuperar objetos de um servidor
- Consultar o servidor para obter informações sobre objetos armazenados
- Gerenciar espaços no arquivo
- Enviar eventos de retenção

Quando você, como um desenvolvedor de aplicativos, instala a API, recebe os arquivos que um usuário final de um aplicativo necessita:
- A biblioteca compartilhada de API.
- O arquivo de mensagens.
- Os arquivos de opções do cliente de amostra.
- O código-fonte para os arquivos de cabeçalho da API que o aplicativo precisa.
- O código-fonte para um aplicativo de amostra e o arquivo de criação para construí-lo.
- O arquivo dsmtca (UNIX e Linux apenas) .

Para aplicativos de 64 bits, todas as compilações devem ser executadas utilizando-se opções do compilador que ativam o suporte a 64 bits. Por exemplo, '-q64' deve ser usado ao construir aplicativos da API no AIX, e '-m64' deve ser usado no Linux. Consulte os arquivos de criação de amostra para obter informações adicionais.

Importante: Ao instalar a API, certifique-se de que todos os arquivos estejam no mesmo nível.

Para obter informações sobre a instalação da API, consulte Instalando os clientes de backup-archive do IBM Spectrum Protect.

As referências ao UNIX e Linux incluem sistemas operacionais AIX, HP-UX, Linux, Mac OS X e Oracle Solaris.

Entendendo os Arquivos de Configuração e Opções

Os arquivos de configuração e opções configuram as condições e os limites sob os quais a sessão é executada.

Você, um administrador ou usuário final, pode configurar valores de opções para:
- Configurar a conexão para um servidor
- Controlar quais objetos são enviados ao servidor e a qual classe de gerenciamento estão associados

Você define opções em um ou dois arquivos quando instala a API em sua estação de trabalho.

Nos sistemas operacionais UNIX e Linux, as opções residem em dois arquivos de opções:
- dsm.opt - o arquivo de opções do cliente
- dsm.sys - o arquivo de opções do sistema do cliente

Em outros sistemas operacionais, o arquivo de opções do cliente (dsm.opt) contém todas as opções.

Restrição: A API não suporta as seguintes opções de cliente de backup-archive:
- `autofsrename`
- `changingretries`
- `domain`
- `eventlogging`
- `groups`
- `subdir`
- `users`
- `virtualmountpoint`

Você também pode especificar opções na chamada de função **dsmInitEx**. Utilize o parâmetro da cadeia de opções ou o parâmetro do arquivo de configuração da API.

A mesma opção pode derivar de mais de uma origem de configuração. Quando isso ocorre, a origem com a prioridade mais alta tem prioridade. O Tabela 1 lista a sequência de prioridades.

Tabela 1. Origens de Configuração na Ordem de Prioridade Decrescente

Prioridade	UNIX e Linux	Windows	Descrição
1	arquivo dsm.sys (opções do sistema do cliente)	não aplicável	Este arquivo contém opções que um administrador do sistema configura somente para UNIX e Linux. **Dica:** Caso o arquivo dsm.sys contenha sub-rotinas do servidor, certifique-se de que a opção `passwordaccess` especifique o mesmo valor (`prompt` ou `generate`) em cada uma das sub-rotinas.
2	Sequência de opções (opções do cliente)	Sequência de opções (todas as opções)	Uma dessas opções tem efeito quando é transmitida como um parâmetro para uma chamada de **dsmInitEx**. A lista pode conter opções do cliente como `compressalways`, `servername` (somente UNIX e Linux) ou `tcpserveraddr` (não UNIX). Com a cadeia de opções da API, um cliente aplicativo pode fazer alterações nos valores das opções no arquivo de configuração da API e no arquivo de opções do cliente. Por exemplo, seu aplicativo pode consultar o usuário final se compactação for requerida. Dependendo das respostas do usuário, é possível construir uma cadeia de opções da API com essa opção e transmiti-la na chamada a **dsmInitEx**. Para obter informações sobre o formato da cadeia de opções da API, consulte "**dsmInitEx**" na página 121. Você também pode configurar esse parâmetro para `NULL`. Isso indica que não há nenhuma cadeia de opções da API para a sessão.

Tabela 1. Origens de Configuração na Ordem de Prioridade Decrescente (continuação)

Prioridade	UNIX e Linux	Windows	Descrição
3	arquivo de configuração da API (opções do cliente)	arquivo de configuração da API (todas as opções)	Os valores definidos no arquivo de configuração da API substituem os valores configurados no arquivo de opções do cliente. Defina as opções no arquivo de configuração da API com valores que sejam adequados na sessão do IBM Spectrum Protect para o usuário. Os valores são efetivados quando o nome do arquivo de configuração da API é transmitido como um parâmetro na chamada **dsmInitEx**. Esse parâmetro também pode ser configurado para NULL. Isso indica que não há nenhum arquivo de configuração da API para a sessão.
4	arquivo dsm.opt (opções do cliente)	arquivo dsm.opt (todas as opções)	Nos sistemas operacionais UNIX e Linux, o arquivo dsm.opt contém apenas as opções do usuário. Em outros sistemas operacionais, o arquivo dsm.opt contém todas as opções. Para substituir as opções nesses arquivos, siga os métodos descritos na tabela.

Conceitos relacionados:

Opções de processamento

Configurando o Ambiente da API

A API utiliza variáveis de ambiente exclusivas para localizar arquivos. Você pode utilizar diferentes arquivos para aplicativos da API entre os quais o cliente de backup/archive utiliza. Os aplicativos podem utilizar a chamada de função **dsmSetup** para substituir os valores configurados pelas variáveis de ambiente.

Dica: No Windows, o diretório de instalação padrão é: `%SystemDrive%\Program Files\Common Files\Tivoli\TSM\api`

A Tabela 2 lista as variáveis de ambiente da API por sistema operacional.

Tabela 2. Variáveis de Ambiente da API

Variáveis	UNIX e Linux	Windows
DSMI_CONFIG	O nome completo do arquivo de opções do cliente (dsm.opt).	O nome completo do arquivo de opções do cliente (dsm.opt).
DSMI_DIR	Aponta para o caminho que contém dsm.sys, dsmtca, o subdiretório en_US e qualquer outro idioma NLS (Suporte ao Idioma Nacional). O subdiretório en_US deve conter o dsmclientV3.cat.	Aponta para o caminho que contém dscenu.txt e qualquer arquivo de mensagem NLS.
DSMI_LOG	Aponta para o caminho do arquivo dsierror.log.	Aponta para o caminho do arquivo dsierror.log. Se a opção **errorlogname** do cliente estiver configurada, o local especificado por essa opção substituirá o diretório especificado por DSMI_LOG.

Capítulo 2. Construindo e Executando o Aplicativo de API de Amostra

O pacote da API inclui aplicativos de amostra que demonstram as chamadas de função da API no contexto. Instale o aplicativo de amostra e revise o código-fonte para entender como é possível utilizar as chamadas de função.

Selecione um dos seguintes pacotes de aplicativos de API de amostra:
- O pacote de aplicativos interativos com encadeamento único (dapi*)
- O pacote de aplicativos multiencadeados (callmt*)
- O aplicativo de teste de agrupamento de objetos lógicos (dsmgrp*)
- O aplicativo de amostra de política de retenção baseado em eventos (callevnt)
- O aplicativo de amostra de exclusão suspensa (callhold)
- O aplicativo de amostra de proteção de retenção de dados (callret)
- O programa de amostra do buffer de dados do IBM Spectrum Protect (callbuff)

Para ajudar você a começar, reveja o procedimento para criar o aplicativo de amostra dapismp para sua plataforma:
- Para aplicativos de UNIX ou Linux, consulte "Arquivos de Origem do Aplicativo de Amostra UNIX ou Linux".
- Para aplicativos Windows, consulte "Aplicativo de amostra Windows de 64 bits" na página 7.

O aplicativo de amostra dapismp cria seus próprios fluxos de dados ao fazer backup ou arquivar objetos. Ele não lê nem grava objetos no sistema de arquivos do disco local. O nome do objeto não corresponde a nenhum arquivo de sua estação de trabalho. A "cadeia de valor inicial" que você emite gera um padrão que pode ser verificado quando o objeto é restaurado ou recuperado. Após a compilação do aplicativo de amostra e da execução de **dapismp** para iniciá-lo, siga as instruções que são exibidas em sua tela.

Arquivos de Origem do Aplicativo de Amostra UNIX ou Linux

Para construir e executar o aplicativo de amostra UNIX ou Linux, é necessário assegurar que você tenha certos arquivos de origem. Depois de criar o aplicativo de amostra, você pode compilá-lo e executá-lo.

Os arquivos listados na Tabela 3 incluem os arquivos de origem e outros arquivos necessários para construir o aplicativo de amostra incluído com o pacote da API.

Tabela 3. Arquivos Necessários para Construir o Aplicativo de Amostra da API do UNIX ou Linux

Nomes de Arquivos	Descrição
README_api_enu	Arquivo LEIA-ME
dsmrc.h	Arquivo de cabeçalho dos códigos de retorno
dsmapitd.h	Arquivo de cabeçalho de definições de tipo comum
dsmapips.h	Arquivo de cabeçalho de definições de tipo específico do sistema operacional
dsmapifp.h	Arquivo de cabeçalho do protótipo da função
release.h	Arquivo de cabeçalho dos valores do release

Tabela 3. Arquivos Necessários para Construir o Aplicativo de Amostra da API do UNIX ou Linux (continuação)

Nomes de Arquivos		Descrição
dapibkup.c dapidata.h dapiinit.c dapint64.h dapint64.c dapipref.c dapiproc.c dapiproc.h	dapipw.c dapiqry.c dapirc.c dapismp.c dapitype.h dapiutil.h dapiutil.c	Módulos para o aplicativo de amostra orientado pela linha de comandos
makesmp[64].*xxx*		Arquivo de criação para construir dapismp para seu sistema operacional. *xxx* indica o sistema operacional.
callmt1.c callmt2.c		Arquivos de amostra multiencadeado
callmtu1.c callmtu2.c		Arquivos de amostra Unicode multiencadeados
libApiDS.xx libApiDS64.xx, ou libApiTSM64.xx		Biblioteca compartilhada (o sufixo depende da plataforma)
dsmgrp.c callevnt.c callhold.c callret.c callbuff.c dpsthread.c		Agrupamento de arquivos de amostra Código-fonte da amostra da política de retenção baseado em eventos Código-fonte da amostra de exclusão suspensa Código-fonte da amostra de proteção de retenção de dados

Criando o Aplicativo de Amostra UNIX ou Linux

Você constrói o aplicativo da API de amostra **dapismp** usando um compilador para seu sistema operacional.

Você deve instalar os compiladores a seguir para construir o aplicativo de amostra da API de UNIX ou Linux:

- IBM AIX - compilador IBM Visual Age Versão 6 ou posterior
- Compilador HP-IA64 - aCC A.05.50 ou posterior
- Linux - Compilador GCC Versão 3.3.3 ou posterior
- Mac OS X - Compilador GCC Versão 4.0 ou posterior
- Oracle Solaris - Oracle Studio C++ compiler Versão 11 ou posterior

1. Para construir as amostras da API, execute o comando a seguir:

   ```
   gmake -f makesmp[64].xxx
   ```

   Onde *xxx* indica o sistema operacional.
2. Depois de criar as amostras, configure suas variáveis de ambiente, incluindo `DSMI_DIR`, e seus arquivos de opções. Para obter informações adicionais, consulte "Entendendo os Arquivos de Configuração e Opções" na página 1.
3. A primeira vez que você efetuar logon, faça isso como o usuário raiz para registrar sua senha.

   Dica: A configuração da opção `compressalways` para `no` pode não reenviar um objeto não-compactado. Esse comportamento depende da funcionalidade do aplicativo.

Para especificar o método de comunicações de Memória Compartilhada em AIX, o usuário cliente da API IBM Spectrum Protect deve estar em conformidade com uma das condições a seguir:

- Deve estar conectado como um usuário raiz.
- Deve ter o mesmo UID como o processo que é executado no servidor IBM Spectrum Protect.

Essa restrição não se aplica se a opção `passwordaccess` for configurada para `generate` no arquivo de opções do sistema do cliente `dsm.sys` e o TCA estiver sendo usado ou se você alterar as permissões do seu arquivo de programa do aplicativo usando os comandos a seguir:

```
chown root.system your_api_program
chown u+s your_api_program
```

Para obter mais informações, consulte a documentação do programa de aplicativo.

4. Execute o comando **dapismp** para iniciar o aplicativo.
5. Escolha na lista de opções que é exibida. Assegure-se de que você execute a ação de conexão antes de executar qualquer outra ação.

Exigência: Sempre prefixe os nomes de alto e baixo nível do espaço no arquivo com o delimitador de caminho correto (/) ao inserir o nome, por exemplo: `/myfilespace`. Você deve usar este prefixo ao especificar o caractere curinga asterisco (*).

Conceitos relacionados:

Variáveis de ambiente (sistemas UNIX e Linux)

Aplicativo de amostra Windows de 64 bits

Para construir e executar o aplicativo de amostra para sistemas Microsoft Windows de 64 bits, você deve instalar a API do IBM Spectrum Protect e certificar-se de possuir determinados arquivos de origem.

Restrições:

- Para obter melhores resultados, use o carregamento dinâmico. Para um exemplo, consulte o arquivo `dynaload.c` e a implementação no código de amostra.
- Arquivos para o aplicativo de amostra estão nos diretórios a seguir:

  api64\obj
  Contém os arquivos do objeto de programa de amostra do API.

  api64\samprun
  Contém o programa de amostra **dapismp**. O programa de amostra contém o diretório de execução.

- A DLL tsmapi64.dll é uma DLL de 64 bits.
- Use o Microsoft C/C++ Compiler Version 15 e o makefile `makesmp64.mak` para compilar o aplicativo de amostra do API **dapismp**. Você pode ter que ajustar os makefiles para se ajustarem a seu ambiente, especificamente à biblioteca ou diretórios de inclusão.
- Depois de compilar o aplicativo, execute o aplicativo de amostra emitindo o comando **dapismp** no diretório `api64\samprun`.
- Escolha na lista de opções que é exibida. Assegure-se de que você execute a ação de conexão antes de executar qualquer outra ação.

- Sempre prefixe os nomes de alto e baixo nível do espaço no arquivo com o delimitador de caminho correto (\) ao inserir o nome, por exemplo: \myfilespace. Você deve usar este prefixo ao especificar o caractere curinga asterisco (*).

Para sistemas operacionais Windows, os arquivos de origem que você deve ter para construir o aplicativo de amostra estão listados em Tabela 4. O aplicativo de amostra está incluído no pacote do API. Por conveniência, um executável pré-compilado (dapismp.exe) também é incluído.

Tabela 4. Arquivos para Construir o Aplicativo de Amostra do API de Windows de 64 Bits

Nomes de Arquivos	Descrição
api.txt	Arquivo LEIA-ME
tsmapi64.dll	DLLs da API
dsmrc.h dsmapitd.h dsmapips.h dsmapifp.h dsmapidl.h release.h	Arquivo de cabeçalho dos códigos de retorno Arquivo de cabeçalho de definições de tipo comum Arquivo de cabeçalho de definições de tipo específico do sistema operacional Arquivo de cabeçalho do protótipo da função Arquivo de cabeçalho do protótipo da função carregada dinamicamente Arquivo de cabeçalho dos valores do release
dapidata.h dapint64.h dapitype.h dapiutil.h	Arquivos de cabeçalho do código de origem
tsmapi64.lib	Biblioteca implícita
dapibkup.c dapiinit.c dapint64.c dapipref.c dapiproc.c dapiproc.h dapipw.c dapiqry.c dapirc.c dapismp64.c dapiutil.c dynaload.c	Arquivos de código-fonte ou dapismp.exe
makesmpx64.mak (Windows x64) makesmp64.mak (Windows IA64)	Arquivos de criação para construir aplicativos de amostra
callmt1.c callmt2.c callmtu164.c callmtu264.c dpsthread.c	Arquivos de amostra multiencadeados Código-fonte do arquivo de amostra
callevnt.c callhold.c callret.c callbuff.c	Código-fonte da política de retenção baseado em eventos Código-fonte da amostra de exclusão suspensa Código-fonte da amostra de proteção de retenção de dados Código-fonte de amostra do buffer compartilhado (nenhuma cópia).

Capítulo 3. Considerações para Projetar um Aplicativo

Ao projetar um aplicativo, você deve ter um amplo entendimento de muitos aspectos da API.

Para obter um entendimento da API, revise os tópicos a seguir:

Ao projetar seu aplicativo, revise as considerações em Tabela 5. Estruturas iniciais com campos **`memset`** podem alterar em liberações subsequentes. O valor `stVersion` incrementa com cada aprimoramento do produto.

Tabela 5. Considerações da API para Projetar um Aplicativo

Item de design	Considerações
Configurando o Código do Idioma	O aplicativo deve configurar o código de idioma antes de a API ser chamada. Para configurar o código de idioma para o valor padrão, inclua o código a seguir no aplicativo: `setlocale(LC_ALL,"");` Para configurar o código de idioma para outro valor, use a mesma chamada com o código de idioma adequado no segundo parâmetro. Verifique se há informações específicas na documentação para cada sistema operacional que você estiver usando.

Tabela 5. Considerações da API para Projetar um Aplicativo (continuação)

Item de design	Considerações

Controle de sessão	Aplique as diretrizes a seguir no controle de sessão:
	• Designe o nome do nó exclusivo para cada cliente de backup-archive do IBM Spectrum Protect e produto do cliente da API do IBM Spectrum Protect que você usar. Os produtos a seguir são exemplos destes clientes:
	– IBM Spectrum Protect for Mail
	– ou IBM Spectrum Protect HSM para Windows
	• Use um nome do proprietário consistente através de um procedimento de backup e restauração.
	• Use a opção `passwordaccess` para gerenciar o acesso ao arquivo de senha protegido. Esta opção afeta o uso do processo-filho do TCA apenas em UNIX e Linux, para o nome do nó, nome do proprietário da sessão e gerenciamento de senha.
	• Assegure-se de que as sessões para a movimentação de dados terminem quando a tarefa estiver concluída de forma que os dispositivos no servidor sejam liberados para uso por outras sessões.
	• Para permitir a transferência de dados sem a LAN, use a chamada de função **dsmSetup** com sinalizador `multiencadeado` configurado como on.
	• Em AIX, quando estiver usando aplicativos multiencadeados ou sem a LAN, especificamente executando em máquinas com diversos processadores, configure a variável de ambiente `AIXTHREAD_SCOPE` para `S` no ambiente antes de iniciar o aplicativo, para obter melhor desempenho e mais planejamento sólido. Por exemplo:
	`EXPORT AIXTHREAD_SCOPE=S`
	Ao configurar `AIXTHREAD_SCOPE` para `S`, encadeamentos de usuários que são criados com atributos padrão são posicionados no escopo de contenção de todo o sistema. Se um encadeamento de usuários for criado com escopo de contenção de todo o sistema, o encadeamento de usuários limitado a um encadeamento kernel e é planejado pelo kernel. O encadeamento kernel subjacente não é compartilhado com nenhum encadeamento de usuário. Para obter mais informações sobre esta variável de ambiente, consulte o tópico a seguir:
	"Usando Multiencadeamentos" na página 15
	• Assegure-se de que apenas um encadeamento em uma sessão chama qualquer função da API a qualquer momento. Os aplicativos que usam diversos encadeamentos com a mesma manipulação de sessão devem sincronizar as chamadas da API. Por exemplo, utilize um **mutex** para sincronizar chamadas da API:
	getTSMMutex()
	emita a chamada da API do TSM
	releaseTSMMutex()
	Use esta abordagem apenas quando os encadeamentos compartilharem uma manipulação. É possível usar chamadas paralelas para funções da API se as chamadas tiverem manipulações de sessão diferentes.
	• Implemente um modelo de consumidor/produtor encadeamento para movimentação de dados. Chamadas da API são síncronas e as chamadas para **dsmGetData function** e **dsmSendData function** são bloqueadas até que sejam concluídas. Ao usar um modelo de consumidor/produtor, o aplicativo pode ler o próximo buffer durante períodos de espera para a rede. Além disso, a separação da leitura/gravação de dados é a rede aumentam o desempenho quando há gargalo ou atrasos na rede. Em geral, permanece o seguinte:
	`Data thread <---> shared queue of buffers <---> communication thread (issue calls to the IBM Spectrum Protect API)`
	• Use a mesma sessão para diversas operações para evitar incorrer em uma sobrecarga. Para aplicativos que lidam com muitos objetos pequenos, implemente o conjunto de sessões de forma que a mesma sessão possa ser usada através de diversas operações pequenas. Uma sobrecarga é associada com a abertura e fechamento de uma sessão para o servidor IBM Spectrum Protect. A chamada `dsmInit/dsmInitEX` é serializada de forma que mesmo em um aplicativo multiencadeado apenas um encadeamento pode se conectar a qualquer momento. Além disso, durante a conexão a API envia várias consultas de uma vez para o servidor de forma que ele possa executar todas as operações. Estas consultas incluem política, opção, espaços no arquivo e configuração local.

Seqüência de operações	O servidor IBM Spectrum Protect bloqueia as entradas do banco de dados de espaço no arquivo durante algumas operações. As regras a seguir se aplicam quando você estiver projetando aplicativos da API do IBM Spectrum Protect: • Consultas bloqueiam o espaço no arquivo durante toda a transação. • O bloqueio da consulta pode ser compartilhado com outras operações de consulta, de forma que várias operações de consulta no mesmo espaço no arquivo podem ser executadas simultaneamente. • As operações a seguir são usadas para modificar o banco de dados do servidor IBM Spectrum Protect (**DB Chg**): enviar, obter, renomear, atualizar e excluir. • A conclusão de uma operação **DB Chg** requer um bloqueio de espaço no arquivo durante a alteração do banco de dados no final da transação. • Várias operações **DB Chg** no mesmo espaço no arquivo podem ser executadas simultaneamente. Pode haver um atraso enquanto a sequência espera pelo bloqueio na transação de término. • O bloqueio da consulta não pode ser compartilhado com operações **DB Chg**. Uma operação **DB Chg** atrasa o início de uma consulta no mesmo espaço no arquivo, portanto projete seus aplicativos para separar e serializar consultas de operações **DB Chg** no mesmo espaço no arquivo.
Nomenclatura do objeto	Ao nomear seus objetos, considere os fatores a seguir: • Os nomes do objeto específico são os nomes do objeto de alto e baixo nível. Se um identificador exclusivo, como um registro de data, for incluído no nome, então, os objetos de backup sempre estarão ativos. Os objetos expiram apenas quando são marcados como inativos pela chamada de função **dsmDeleteObj**. • O método de restauração para objetos determina como formar o nome para consultas fáceis. Se você planeja utilizar uma POR (Restauração Parcial do Objeto), você não pode utilizar compactação. Para comprimir a compactação, use a função **dsmSendObj objAttr objCompressed=bTrue**.
Agrupamento do objeto	Agrupe objetos logicamente usando espaços no arquivo. Um espaço no arquivo é um contêiner no servidor que fornece uma categoria de agrupamento para os objetos. A API consulta todos os espaços no arquivo durante a conexão inicial e também durante as consultas, de forma que o número de espaços no arquivo deve ser restringido. Uma suposição razoável é que um aplicativo configura até 20 a 100 espaços de arquivo por nó. A API pode atender para mais arquivos de espaços, mas cada espaço no arquivo incorre em uma sobrecarga para a sessão. Para criar uma separação granular, use o objeto `directory` no aplicativo.

Tabela 5. Considerações da API para Projetar um Aplicativo (continuação)

Item de design	Considerações
Manipulação de Objeto	Não armazene valores `objectID` para uso para restaurações futuras. Não há garantias de que estes valores sejam persistentes durante a vida do objeto. Durante uma restauração, preste atenção especial à ordem de restauração. Depois da consulta, classifique neste valor antes da restauração. Se estiver usando diversos tipos de mídia serial, então acesse os tipos diferentes de mídia em sessões separadas. Para obter informações adicionais, consulte o tópico a seguir: "Selecionando e Classificando Objetos pela Ordem de Restauração" na página 69
Classe de gerenciamento	Considere quanto controle o aplicativo deve ter sobre a classe de gerenciamento que está associada com os objetos de aplicativo. Você pode definir instruções de inclusão ou especificar um nome na chamada de função **`dsmSendObj`**.
Tamanho do Objeto	O IBM Spectrum Protect precisa saber uma estimativa de tamanho para cada objeto. Considere como seu aplicativo estima o tamanho de um objeto. Uma superestimação do tamanho do objeto é melhor que uma subestimação.

Determinando Limites de Tamanho

Determinadas estruturas de dados ou campos na API têm limites de tamanho. Essas estruturas são freqüentemente nomes ou outros campos de texto que não podem exceder um comprimento predeterminado.

Os campos a seguir são exemplos de estruturas de dados que possuem limites de tamanho:
- Tipo de aplicativo
- Descrição de archive
- Destino do grupo de cópias
- Nome do grupo de cópias
- Informações do espaço no arquivo
- Nome da classe de gerenciamento
- Nome do proprietário do objeto
- Password

Estes limites são definidos como constantes no arquivo de cabeçalho `dsmapitd.h`. Qualquer alocação de armazenamento é baseada nestas constantes ao invés de em números inseridos. Para obter informações adicionais, consulte Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163.

Mantendo o Controle de Versão da API

Todas as APIs têm alguma forma de controle de versão. A versão da API utilizada no aplicativo deve ser compatível com a versão da biblioteca de API instalada na estação de trabalho do usuário.

`dsmQueryApiVersionEx` deve ser a primeira chamada da API digitada ao utilizar a API. Esta chamada executa as tarefas a seguir:
- Confirma que a biblioteca da API está instalada e disponível no sistema do usuário final
- Retorna o nível de versão da biblioteca da API que o aplicativo acessa

A API é projetada para ser compatível com futuras versões. Os aplicativos que são gravados em versões ou liberações anteriores da biblioteca da API funcionam corretamente quando uma versão mais recente é executada.

A determinação do release da biblioteca da API é muito importante, pois alguns releases podem ter diferentes requisitos de memória e definições de estrutura de dados. A retrocompatibilidade é improvável. Consulte a Tabela 6 na página 14 para obter informações sobre sua plataforma.

Tabela 6. Informações de Compatibilidade de Plataforma

Plataforma	Descrição
Windows	Os arquivos de mensagens devem estar no mesmo nível que a biblioteca (DLL). O módulo do Agente de Comunicação Confiável (dsmtca) não é usado.
UNIX ou Linux	A biblioteca da API o módulo do Agente de Comunicação Confiável (dsmtca) e os arquivos de mensagens devem estar no mesmo nível.

A chamada **dsmQueryApiVersionEx** retorna a versão da biblioteca da API instalada na estação de trabalho do usuário final. Em seguida, é possível comparar o valor retornado à versão da API que o cliente aplicativo está utilizando.

O número da versão da API do aplicativo cliente é digitado no código do objeto compilado como um conjunto de quatro constantes definidas em dsmapitd.h:

```
DSM_API_VERSION
DSM_API_RELEASE
DSM_API_LEVEL
DSM_API_SUB_LEVEL
```

Consulte a seção Apêndice B, “Arquivos de Origem de Definições de Tipos de API”, na página 163.

A versão da API do cliente aplicativo deve ser menor ou igual a da biblioteca da API instalada no sistema do usuário. Tome cuidado com qualquer outra condição. É possível digitar a chamada **dsmQueryApiVersionEx** a qualquer momento, tenha a sessão da API sido iniciada ou não.

As estruturas de dados utilizadas pela API também têm informações de controle. As estruturas têm informações de versão no primeiro campo. À medida que aprimoramentos são feitos nas estruturas, o número da versão é aumentado. Ao inicializar o campo da versão, utilize o valor da Versão da estrutura definido em dsmapitd.h.

Figura 1 na página 15 demonstra a definição do tipo da estrutura, **dsmApiVersionEx** do arquivo de cabeçalho, dsmapitd.h. O exemplo define então uma variável global denominada ***apiLibVer***. Também demonstra como é possível utilizá-la em uma chamada **dsmQueryApiVersionEx** para retornar a versão da biblioteca da API do usuário final. Por fim, o valor retornado é comparado ao número da versão da API do cliente aplicativo.

```
typedef struct
{
        dsUint16_t     stVersion;           /* versão da estrutura             */
        dsUint16_t version;       /* Versão da API                    */
        dsUint16_t release;       /* Release da API                   */
        dsUint16_t level;         /* Nível da API                     */
        dsUint16_t subLevel;      /* Subnível da API                  */
} dsmApiVersionEx;

dsmApiVersionEx apiLibVer;

memset(&apiLibVer,0x00,sizeof(dsmApiVersionEx));
dsmQueryApiVersionEx(&apiLibVer);

/* verificar problemas de compatibilidade */
dsInt16_t appVersion= 0, libVersion = 0;
  appVersion=(DSM_API_VERSION * 10000)+(DSM_API_RELEASE * 1000) +
              (DSM_API_LEVEL * 100) + (DSM_API_SUBLEVEL);
  libVersion = (apiLibVer.version * 10000) + (apiLibVer.release * 1000) +
                  (apiLibVer.level * 100) + (apiLibVer.subLevel);
   if (libVersion < appVersion)
   {
      printf("\n************************************************************\n");
      printf("The IBM Spectrum Protect API library is lower than the application version\n");
      printf("Install the current library version.\n");
      printf("************************************************************\n");
      return 0;
   }

printf("* API Library Version = %d.%d.%d.%d  *\n",
     apiLibVer.version,
     apiLibVer.release,
     apiLibVer.level,
     apiLibVer.subLevel);
```

Figura 1. Um Exemplo de Obtenção do Nível da Versão da API

Usando Multiencadeamentos

Uma API multiencadeada permite que os aplicativos criem várias sessões com o servidor IBM Spectrum Protect no mesmo processo. A API pode ser digitada novamente. As chamadas executadas em paralelo a partir de diferentes encadeamentos.

Dica: Quando você executar aplicativos que assumam uma API multiencadeada, utilize a chamada **dsmQueryAPIVersionEx**.

Para executar a API no modo multiencadeado, configure o valor de *mtflag* para DSM_MULTITHREAD na chamada de **dsmSetUp**. A chamada de **dsmSetUp** deve ser a primeira chamada após a chamada de **dsmQueryAPIVersionEx**. Essa chamada deve ser retornada antes de qualquer encadeamento fazer a chamada de **dsmInitEx**. Quando todos os encadeamentos concluírem o processamento, digite uma chamada para **dsmCleanUp**. O processo primário não deve terminar antes de todos os encadeamentos concluírem o processamento. Consulte callmt1.c no aplicativo de amostra.

Restrição: O padrão para a API é o modo de encadeamento único. Se um aplicativo não chamar **dsmSetUp** com o valor de *mtflag* configurado para DSM_MULTITHREAD, a API permite somente uma sessão para cada processo.

Quando **dsmSetUp** for concluída com sucesso, o aplicativo pode iniciar vários encadeamentos e digitar várias chamadas de **dsmInitEx**. Cada chamada de **dsmInitEx** retorna um identificador para a sessão. As chamadas subseqüentes nesse encadeamento para a sessão devem utilizar o valor do identificador. Determinados valores são variáveis de ambiente para todo o processo (valores configurados em **dsmSetUp**). Cada chamada de **dsmInitEx** analisa as opções novamente. Cada

encadeamento pode ser executado com diferentes opções, especificando-se um arquivo de sobrescrita ou uma cadeia de opções na chamada de **dsmInitEx**. Isso permite que diferentes encadeamentos vão para diferentes servidores ou utilizam diferentes nomes de nós.

Recomendação: No HP, configure a pilha de encadeamentos para 64 K ou mais. O valor padrão da pilha de encadeamentos (32 K) pode não ser suficiente.

Para permitir que os usuários de aplicativos tenham uma sessão sem a LAN, utilize **dsmSetUp** *mtFlag DSM_MULTITHREAD* no seu aplicativo. Isso é necessário mesmo se o aplicativo tiver encadeamento único. Esse sinalizador ativará o encadeamento necessário para a interface sem a LAN do IBM Spectrum Protect.

Sinais e Manipuladores de Sinais

O aplicativo manipula sinais do usuário ou do sistema operacional. Se o usuário inserir uma sequência de pressionamento de tecla **CTRL+C**, o aplicativo deve capturar o sinal e enviar chamadas **dsmTerminate** para cada um dos encadeamentos ativos. Em seguida, chamar **dsmCleanUp** para sair. Se as sessões não forem encerradas adequadamente, resultados inesperados podem ocorrer no servidor.

O aplicativo requer manipuladores de sinais, como SIGPIPE e SIGUSR1, para sinais que fazem o aplicativo terminar. O aplicativo então recebe o código de retorno da API. Por exemplo, para ignorar inclua SIGPIPE e a instrução a seguir em seu aplicativo: `signal(SIGPIPE, SIG_IGN)`. Depois que estas informações forem incluídas, ao invés do aplicativo existente em um canal de quebra, o código de retorno quebra é retornado.

Você pode usar o processo-filho, TCA (Agente de Comunicação Confiável), se a opção `passwordaccess` for configurada como `generate`. Quando o processo de TCA é usado, o IBM Spectrum Protect usa o sinal SIGCLD. Caso seu aplicativo use o sinal SIGCLD, esteja ciente da possível interferência dos processos do IBM Spectrum Protect e sobre como SIGCLD é utilizado. Para obter mais informações sobre como usar o TCA, consulte “Segurança da Sessão” na página 18.

Iniciando ou Encerrando uma Sessão

IBM Spectrum Protect é um produto baseado em sessão e todas as atividades devem ser executadas dentro de uma sessão do IBM Spectrum Protect. Para iniciar uma sessão, o aplicativo inicia a chamada de **dsmInitEx**. Essa chamada deve ser executada antes de qualquer outra chamada da API diferente de **dsmQueryApiVersionEx**, **dsmQueryCliOptions** ou **dsmSetUp**.

A função **dsmQueryCliOptions** pode ser chamada apenas antes da chamada **dsmInitEx**. A função retorna os valores de opções importantes, como arquivos de opções, configurações de compressão e parâmetros de comunicação. A chamada **dsmInitEx** configura uma sessão com o servidor como indicado nos parâmetros que são passados na chamada ou definidos nos arquivos de opções.

O nome do nó cliente, o nome do proprietário e os parâmetros de senha são transmitidos para a chamada de **dsmInitEx**. O nome do proprietário faz distinção entre maiúsculas e minúsculas, mas o nome do nó e a senha não. Os nós do aplicativo cliente devem ser registrados com o servidor antes de uma sessão iniciar.

Toda vez que um cliente aplicativo da API inicia uma sessão com o servidor, o tipo de aplicativo cliente é registrado no servidor. Sempre especifique uma abreviação

do sistema operacional para o valor de tipo de aplicativo porque este valor é inserido no campo da plataforma no servidor. O comprimento máximo de cadeia é DSM_MAX_PLATFORM_LENGTH.

A chamada de função **dsmInitEx** estabelece a sessão do IBM Spectrum Protect com o arquivo de configuração da API e a lista de opções do cliente aplicativo. O cliente aplicativo pode utilizar o arquivo de configuração da API e a lista de opções para configurar várias opções do IBM Spectrum Protect. Estes valores substituem os valores que são configurados nos arquivos de configuração de usuário durante a instalação. Os usuários não podem mudar as opções definidas pelo administrador. Se o aplicativo cliente não tiver um arquivo de configuração e uma lista de opções específicos, é possível configurar ambos os parâmetros para NULL. Para obter mais informações sobre arquivos de configuração, consulte o tópico a seguir:

"Entendendo os Arquivos de Configuração e Opções" na página 1

A chamada de função **dsmInitEx** estabelece a sessão IBM Spectrum Protect usando os parâmetros que permitem verificação ampliada.

Verifique a chamada de função **dsmInitEx** e o código de retorno de informações de **dsmInitExOut**. O usuário cancelou a última sessão se o código de retorno for okay (RC=ok) e o código de retorno de informações (infoRC) for DSM_RC_REJECT_LASTSESS_CANCELED. Para encerrar a sessão atual imediatamente, chame **dsmTerminate**.

A chamada de **dsmQuerySessOptions** retorna os mesmos campos que a chamada de **dsmQueryCliOptions**. A chamada pode ser enviada apenas em uma sessão. Os valores refletem as opções do cliente válidas durante a sessão de arquivos de opções e de qualquer substituição da chamada de **dsmInitEx**.

Depois de uma sessão iniciar, o aplicativo pode enviar uma chamada para **dsmQuerySessInfo** para determinar os parâmetros do servidor que são configurados para esta sessão. Itens como o domínio de política e os limites de transação são retornados ao aplicativo com a chamada.

Encerre sessões com uma chamada de **dsmTerminate**. Qualquer conexão com o servidor é encerrada e todos os recursos que estão associadas a ela são liberados.

Para obter um exemplo de início e término de sessão, consulte o tópico a seguir:

Figura 2 na página 20

O exemplo define um número de variáveis globais e locais que são usadas em chamadas para **dsmInitEx** e **dsmTerminate**. A chamada **dsmInitEx** toma um ponteiro para `dsmHandle` como um parâmetro, enquanto a chamada **dsmTerminate** toma o `dsmHandle` como um parâmetro. O exemplo na Figura 3 na página 20 exibe os detalhes de **rcApiOut**. A função **rcApiOut** chama a função do API **dsmRCMsg**, que traduz um código de retorno em uma mensagem. A chamada de **rcApiOut**, em seguida, imprime a mensagem para o usuário. Uma versão de **rcApiOut** é incluída no aplicativo de amostra da API. A função **dsmApiVersion** é um tipo de definição que é localizada no arquivo de cabeçalho `dsmapitd.h`.

Segurança da Sessão

O IBM Spectrum Protect, um sistema baseado em sessão, possui componentes de segurança que permitem que os aplicativos iniciem sessões de maneira segura. Essas medidas de segurança proíbem o acesso não autorizado ao servidor e ajudam a assegurar a integridade do sistema.

Cada sessão iniciada com o servidor deve concluir um processo de conexão e requer uma senha. Quando a senha forma um par com o nome do nó do cliente, ela assegura autorização apropriada ao se conectar ao servidor. O cliente aplicativo fornece essa senha à API para iniciar a sessão.

Dois métodos de processamento de senha estão disponíveis: *passwordaccess=prompt* ou *passwordaccess=generate*. Se você utilizar a opção *passwordaccess=prompt*, deverá incluir o valor da senha em cada chamada de **dsmInitEx**. Outra opção é fornecer o nome do nó e o nome do proprietário na chamada de **dsmInitEx**.

As senhas têm tempos de expiração associados a elas. Se uma chamada de **dsmInitEx** falhar com um código de retorno de senha expirada (DSM_RC_REJECT_VERIFIER_EXPIRED), o cliente aplicativo deve digitar a chamada de **dsmChangePW** utilizando o identificador retornado por **dsmInitEx**. Isso atualizará a senha antes de sessão poder ser estabelecida com êxito. O exemplo na Figura 4 na página 21 demonstra o procedimento para alterar uma senha, utilizando **dsmChangePW**. O proprietário do login deve ser um ID de usuário raiz ou um ID de usuário autorizado a alterar a senha.

O segundo método, *passwordaccess=generate*, criptografa e armazena o valor da senha em um arquivo. O nome do nó e o nome do proprietário não podem ser fornecidos na chamada de **dsmInitEx** e os valores padrão do sistema serão utilizados. Isso protege a segurança do arquivo de senha. Quando a senha expira, o parâmetro `generate` cria uma nova e atualiza o arquivo de senha automaticamente.

Dicas:

1. Se duas máquinas físicas diferentes tiverem o mesmo nome de nó do IBM Spectrum Protect ou se vários caminhos estiverem definidos em um nó utilizando várias sub-rotinas do servidor, *passwordaccess=generate* poderá funcionar somente para a sub-rotina que é utilizada após a expiração da senha. Durante o primeiro contato cliente-servidor, é solicitada ao usuário a mesma senha para cada sub-rotina do servidor separadamente e, para cada instância, uma cópia da senha é armazenada separadamente. Quando a senha expira, uma nova senha é gerada para a sub-rotina que conecta o primeiro contato cliente-servidor. Todas as tentativas subsequentes para conexão através de outras sub-rotinas do servidor falharão, pois não há nenhum link lógico entre as respectivas cópias da senha antiga e a cópia atualizada gerada pela primeira sub-rotina utilizada após a expiração da senha. Nesse caso, você deve atualizar as senhas antes da expiração ou após a expiração como uma recuperação da situação, da seguinte forma:
   a. Execute **dsmadmc** e atualize a senha no servidor.
   b. Execute **dsmc -servername=stanza1** e utilize a nova senha para gerar uma entrada apropriada.
   c. Execute **dsmc -servername=stanza2** e utilize a nova senha para gerar uma entrada apropriada.

2. Para UNIX ou Linux: somente o usuário raiz ou um usuário autorizado pode alterar a senha ao utilizar *passwordaccess=prompt*. Somente o usuário raiz ou um usuário autorizado pode iniciar o arquivo de senha ao utilizar *passwordaccess=generate*. É possível utilizar o processo-filho do TCA (Agente de Comunicação Confiável) para processar a senha. O aplicativo deve estar ciente disso, pois um processo-filho e o sinal SIGCLD são utilizados. O TCA não é utilizado nestas situações:
   - A opção *passwordaccess* é configurada para *prompt*.
   - O usuário de login é root.
   - O responsável pela chamada da função deve ser um usuário autorizado.

   Restrição: As opções `users` e `groups` não são reconhecidas.

Um aplicativo pode restringir o acesso do usuário por outros meios, como configurando filtros de acesso.

Os aplicativos que utilizam várias conexões de IP para um único servidor IBM Spectrum Protect devem usar o mesmo nome de nó e senha do cliente IBM Spectrum Protect para cada sessão. Siga estas etapas para ativar esse suporte:

1. Defina uma sub-rotina do servidor IBM Spectrum Protect no arquivo dsm.sys.
2. Para as conexões que não estão utilizando o endereço IP padrão, especifique os valores de opção para o endereço *TCPserver* e *TCPport* na chamada de **dsmInitEx**.

Esses valores irão substituir as informações de conexão de IP, mas a sessão ainda utilizará as mesmas informações de nó e senha da sub-rotina de dsm.sys.

Nota: Os nós de um cluster compartilham uma única senha.

```
dsmApiVersionEx * apiApplVer;
char           *node;
char          *owner;
char           *pw;
char           *confFile = NULL;
char           *options = NULL;
dsInt16_t      rc = 0;
dsUint32_t     dsmHandle;
dsmInitExIn_t  initIn;
dsmInitExOut_t initOut;
char           *userName;
char           *userNamePswd;

memset(&initIn, 0x00, sizeof(dsmInitExIn_t));
memset(&initOut, 0x00, sizeof(dsmInitExOut_t));
memset(&apiApplVer,0x00,sizeof(dsmapiVersionEx));
apiApplVer.version = DSM_API_VERSION; /* Configure a versão do tempo de compilação */
apiApplVer.release = DSM_API_RELEASE;  /* time version.         */
apiApplVer.level   = DSM_API_LEVEL;
apiApplVer.subLevel= DSM_API_SUBLEVEL;

printf("Doing signon for node %s, owner %s, with password %s\n", node,owner,pw);

initIn.stVersion = dsmInitExInVersion;
initIn.dsmApiVersionP = &apiApplVer
initIn.clientNodeNameP = node;
initIn.clientOwnerNameP = owner ;
initIn.clientPasswordP = pw;
initIn.applicationTypeP = "Sample-API AIX";
initIn.configfile = confFile;
initIn.options = options;
initIn.userNameP = userName;
initIn.userPasswordP = userNamePswd;
rc = dsmInitEx(&dsmHandle, &initIn, &initOut);

if (rc == DSM_RC_REJECT_VERIFIER_EXPIRED)
{
   printf("*** Senha expirada. Selecione Alterar Senha.\n");
   return(rc);
}
else if (rc)
{
   printf("*** Init failed: ");
   rcApiOut(dsmHandle, rc);   /* Chame função para imprimir mensagem de erro */
   dsmTerminate(dsmHandle);    /* limpe os blocos de memória */
   return(rc);
}
```

Figura 2. Um Exemplo para Iniciar e Encerrar uma Sessão

```
void rcApiOut (dsUint32_t handle, dsInt16_t rc)
{
     char *msgBuf ;

     if ((msgBuf = (char *)malloc(DSM_MAX_RC_MSG_LENGTH+1)) == NULL)
     {
          printf("Abort:  Not enough memory.\n") ;
          exit(1) ;
      }

      dsmRCMsg(handle, rc, msgBuf);
      printf("
      free(msgBuf) ;
      return;
}
```

Figura 3. Detalhes de rcApiOut

```
printf("Enter your current password:");
gets(current_pw);
printf("Enter your new password:");
gets(new_pw1);
printf("Enter your new password again:");
gets(new_pw2);
/* Se as entradas da nova senha não corresponderem, tente novamente ou saia. */
/* Se corresponderem, chame dsmChangePW.                                      */

rc = dsmChangePW(dsmHandle,current_pw,new_pw1);
if (rc)
{
   printf("*** Falha na alteração de senha.  Rc =
}
else
{
   printf("*** Sua nova senha foi aceita e atualizada.\n");
}
return 0;
```

Figura 4. Um Exemplo de Alteração de Senha

Configurando a Opção `passwordaccess` para `generate` Sem TCA

O TCA (Agente de Comunicação Confiável) é um processo-filho que normalmente controla o acesso ao arquivo de senha protegido. Em sistemas UNIX e Linux, é possível efetuar logon como um usuário autorizado e configurar a opção `passwordaccess` como `generate` sem iniciar o TCA.

Conclua as seguintes etapas ao configurar `passwordaccess` como `generate` sem o TCA:

1. Grave o aplicativo com uma chamada para **dsmSetUp**, que transmite *argv[0]*. *argv[0]* contém o nome do aplicativo que chama a API. O aplicativo tem permissão para executar um usuário autorizado; no entanto, o administrador deve decidir o nome de login para o usuário autorizado.
2. Configure o bit do ID de usuário efetivo (bit `S`) para o executável do aplicativo para `On`. O proprietário do arquivo executável do aplicativo pode, então, se tornar um usuário autorizado e pode criar um arquivo de senha, atualizar senhas e executar aplicativos. O proprietário do arquivo executável do aplicativo deve ser igual ao ID do usuário que executa o programa. No exemplo a seguir, *User* é `user1`, o nome do arquivo executável do aplicativo é `applA`, e `user1` tem permissões de leitura/gravação no diretório `/home/user1`. O arquivo executável `applA` possui as seguintes permissões:

   ```
   -rwsr-xr-x user1    group1   applA
   ```
3. Instrua os usuários do aplicativo para que utilizem o nome de usuário autorizado para efetuar login. O IBM Spectrum Protect verifica se o ID de login corresponde ao proprietário do executável do aplicativo antes que permita acesso ao arquivo de senha protegido.
4. Configure a opção `passworddir` no arquivo `dsm.sys` para apontar para um diretório onde este usuário tem acesso de leitura/gravação. Por exemplo, insira a seguinte linha na sub-rotina do servidor do arquivo `dsm.sys`:

   ```
   passworddir /home/user1
   ```
5. Crie o arquivo de senha e certifique-se de que o usuário autorizado seja proprietário do arquivo.
6. Efetue logon como `user1` e execute `app1A`.
7. Chame **dsmSetUp** e transmita *argv*.

Criando um Usuário Administrativo com Autoridade de Proprietário do Cliente

Um usuário administrativo com a autoridade do proprietário cliente pode configurar parâmetros na chamada de função **dsmInitEx** para iniciar sessões. Esse usuário pode funcionar como um "usuário administrativo" com autoridade de backup e restauração para os nós definidos.

Para receber autoridade do proprietário cliente, complete as seguintes etapas no servidor:

1. Defina o usuário administrativo:

   ```
   REGister Admin admin_name password
   ```

   Em que:
   - *admin_name* é o nome do usuário administrativo.
   - *password* é a senha administrativa.
2. Defina o nível de autoridade. Os usuários com autoridade de sistema ou política também têm autoridade de proprietário cliente.

   ```
   Grant Authority admin_name classes authority node
   ```

   Em que:
   - *admin_name* é o usuário administrativo.
   - *classes* é o nó.
   - *authority* tem um dos seguintes níveis de autoridade:
     - `owner`: autoridade de backup completo e restauração para o nó
     - `node`: nó único
     - `domain`: grupo de nós
3. Defina o acesso a um único nó.

   ```
   Registrar o Nó node_name senha userid=user_id
   ```

   Em que:
   - *node_name* é o nó do usuário cliente
   - *password* é a senha do nó do usuário cliente
   - *user_id* é o nome do usuário administrativo

Quando o aplicativo usa o usuário administrativo, a função **dsmInitEx** é chamada com os parâmetros `userName` e `userNamePswd`.

```
dsmInitEx
      clientNodeName = NULL
      clientOwnerName = NULL
      clientPassword = NULL
      userName = 'administrative user' name
      userNamePswd = 'administrative user' password
```

É possível configurar a opção `passwordaccess` como `generate` ou `prompt`. Com qualquer um dos dois parâmetros, o valor de `userNamePswd` inicia a sessão. Quando a sessão iniciar, qualquer processo de backup ou restauração pode ocorrer para esse nó.

IDs e Nomes de Objetos

O servidor IBM Spectrum Protect é um servidor de armazenamento de objetos cuja função primária é armazenar e recuperar eficientemente objetos nomeados. O ID do objeto é exclusivo para cada objeto e permanece pela duração do objeto, *exceto* quando você utiliza exportação ou importação.

Para atender a estes requisitos, o IBM Spectrum Protect tem duas áreas de armazenamento principais, banco de dados e armazenamento de dados.

- O banco de dados contém todos os metadados, como nome ou atributos, associados a um objeto.
- O armazenamento de dados contém os dados do objeto. O armazenamento de dados é na verdade uma hierarquia de armazenamento que o administrador do sistema define. Os dados são armazenados e gerenciados de forma eficiente na mídia on-line ou off-line, dependendo das necessidades de custo e de acesso.

Cada objeto armazenado no servidor tem um nome associado a ele. O cliente controla os seguintes componentes principais com esse nome:

- Nome do Espaço no Arquivo
- Nome de nível superior
- Nome de nível inferior
- Tipo do objeto

Ao tomar decisões sobre a nomenclatura de objetos de um aplicativo, você pode precisar utilizar um nome externo para os nomes completos de objetos para o usuário final. Especificamente, o usuário final pode precisar especificar o objeto em uma instrução de inclusão ou exclusão quando o aplicativo é executado. A sintaxe exata do nome do objeto nessas instruções é dependente da plataforma. No sistema operacional Windows, a letra da unidade associada ao espaço no arquivo em vez de o próprio nome do espaço no arquivo é utilizada na instrução de inclusão ou exclusão.

O valor do ID do objeto que foi designado quando você criou o objeto pode não ser igual ao de quando você executa um processo de restauração. Os aplicativos devem salvar o nome do objeto e, em seguida, consultar para obter o ID do objeto atual antes de uma restauração.

Nome do Espaço no Arquivo

O nome do espaço no arquivo é um dos componentes de armazenamento mais importantes. Ele pode ser o nome de um sistema de arquivos, unidade de disco ou qualquer outro qualificador de nível superior que agrupe dados relacionados.

IBM Spectrum Protect utiliza o espaço no arquivo para identificar o sistema de arquivos ou a unidade de disco na qual os dados estão localizados. Dessa forma, as ações podem ser executadas em todas as entidades de um espaço no arquivo, como consultar todos os objetos de um espaço no arquivo especificado. Como o espaço no arquivo é um componente muito importante da convenção de nomenclatura do IBM Spectrum Protect, utilize chamadas especiais para registrar, atualizar, consultar e excluir espaços no arquivo.

O servidor também tem comandos administrativos para consultar os espaços no arquivo em qualquer nó no armazenamento do IBM Spectrum Protect e excluí-los, se necessário. Todos os dados armazenados pelo cliente aplicativo devem ter um

nome de espaço no arquivo associado a eles. Selecione o nome cuidadosamente para agrupar dados semelhantes no sistema.

Para evitar uma possível interferência, um cliente aplicativo deve selecionar nomes de espaço no arquivo diferentes dos utilizados pelo cliente de backup/archive utilizaria. O aplicativo cliente deve publicar seus nomes de espaços no arquivo de forma que os usuários finais possam identificar os objetos para instruções de inclusão e exclusão, se necessário.

Nota: Nas plataformas Windows, uma letra de unidade é associada a um espaço no arquivo. Ao registrar ou atualizar um espaço no arquivo, você deve fornecer a letra da unidade. Como a lista de inclusão e exclusão refere-se à letra da unidade, você deve monitorar cada letra e seu espaço no arquivo associado. No programa de amostra dapismp, a letra da unidade é configurada para "G" por padrão.

Consulte Capítulo 2, “Construindo e Executando o Aplicativo de API de Amostra”, na página 5 para obter mais informações sobre os programas de amostra.

Nomes de Nível Superior e de Nível Inferior

Dois outros componentes do nome do objeto são o qualificador do nome de nível superior e o qualificador do nome de nível inferior. O qualificador do nome de nível superior é o caminho do diretório ao qual o objeto pertence e o qualificador do nome nível inferior é o nome real do objeto nesse caminho de diretório.

Quando o nome do espaço no arquivo, o nome de nível superior e o nome de nível inferior estão concatenados, eles devem formar um nome sintaticamente correto no sistema operacional no qual o cliente é executado. Não é necessário que o nome exista como um objeto no sistema ou seja semelhante ao dados reais no sistema de arquivos local. No entanto, o nome deve atender às regras de nomenclatura padrão para ser processado corretamente pelas chamadas de **dsmBindMC**. Consulte “Entendendo Objetos de Backup e Archive” na página 43 para obter as considerações sobre nomenclatura relacionadas ao gerenciamento de política.

Tipo do objeto

O tipo de objeto identifica o objeto como um arquivo ou um diretório. Um arquivo é um objeto que contém atributos e dados binários e um diretório é um objeto que contém somente atributos.

Tabela 7 mostra o que o aplicativo cliente codificaria para nomes de objetos pela plataforma.

Tabela 7. Exemplos de Nomes de Objetos de Aplicativos pela Plataforma

Plataforma	Código do Cliente para Nome de Objeto
UNIX ou Linux	/myfs/highlev/lowlev
Windows	"myvol\\highlev\\lowlev" **Nota:** Em uma plataforma Windows, uma barra dupla invertida se converte em uma barra invertida simples, pois uma barra invertida é o caractere de escape. Os nomes de espaços no arquivo começam com uma barra na plataforma UNIX ou Linux, mas não começam com uma barra na plataforma Windows.

Acessando Objetos como Proprietário da Sessão

Cada objeto tem um nome de proprietário associado a ele. As regras que determinam quais objetos são acessados dependem do nome do proprietário utilizado quando uma sessão é iniciada. Utilize o valor de proprietário dessa sessão para controlar o acesso ao objeto.

O proprietário da sessão é configurado durante a chamada a **dsmInitEx** no parâmetro *clientOwnerNameP*. Ao iniciar uma sessão com o nome do proprietário de **dsmInitEx** igual a *NULL* e utilizar *passwordaccess=prompt*, esse proprietário de sessão será manipulado com a autoridade de sessão (raiz ou autorizada pelo usuário). Isso também ocorrerá se você efetuar login com um ID de usuário raiz ou um ID de usuário autorizado e utilizar *passwordaccess= generate*. Durante a sessão iniciada desta maneira, é possível executar qualquer ação em qualquer objeto que seja de propriedade deste nó, independente do proprietário real deste objeto.

Se uma sessão for iniciada com um nome de proprietário específico, a sessão somente poderá executar ações em objetos que tenham esse nome de proprietário de objeto associado a elas. Backups ou archives no sistema devem ter esse nome de proprietário associado a eles. Qualquer consulta executada retorna somente os valores que têm esse nome de proprietário associado ela. O valor do proprietário do objeto é configurado durante a chamada de **dsmSendObj** no campo **Proprietário** da estrutura **ObjAttr**. Um nome de proprietário faz a distinção entre maiúsculas e minúsculas. A Tabela 8 resume as condições sob as quais um usuário tem acesso a um objeto.

Tabela 8. Resumo de Acesso do Usuário aos Objetos

Proprietário da sessão	Proprietário do objeto	Acesso do usuário
NULL (root, proprietário do sistema)	" " (sequência vazia)	Sim
NULL	Nome específico	Yes
Nome específico	" " (sequência vazia)	Não
Nome específico	Mesmo nome	Yes
Nome específico	Nome diferente	Não

Acessando Objetos Através de Nós e Proprietários

Três chamadas de função suportam acesso entre nós e entre proprietários na mesma plataforma: **dsmSetAccess**, **dsmDeleteAccess**, e **dsmQueryAccess**. Essas funções, juntamente com as opções de cadeia *-fromnode* e *-fromowner* que são transmitidas para **dsmInitEx**, permitem um processo de consulta, restauração e recuperação completo entre nós através da API.

Por exemplo, o Usuário A no nó A utiliza a chamada de função **dsmSetAccess** para fornecer acesso a seus backups sob o espaço no arquivo /db ao Usuário B do Nó B. A regra de acesso é exibida como:

ID Alternativo	Tipo	Nó	Usuário	Caminho
1	Cópia de Segurança	Nó B	Usuário B	/db/*/*

Quando o Usuário B efetua logon no Nó B, a cadeia de opção para **dsmInitEx** é:

```
-fromnode=nodeA -fromowner=userA
```

Essas opções são configuradas para essa sessão. As consultas acessarão os espaços no arquivo e os arquivos do Nó A. Backups e archives não são permitidos. Somente os processos de consulta, restauração e recuperação são permitidos a partir dos espaços no arquivo para o qual o Usuário B pode acessar. Se o aplicativo tentar executar uma operação utilizando **dsmBeginTxn** (por exemplo, backup ou atualização) enquanto conectado com uma opção *-fromnode* ou *-fromowner* configurada, **dsmBeginTxn** falhará com o código de retorno DSM_RC_ABORT_NODE_NOT_AUTHORIZED. Consulte as chamadas de função individuais e "**dsmInitEx**" na página 121 para obter informações adicionais.

Dica: No UNIX e no Linux é possível especificar *–fromowner=root* na cadeia de opções que é transmitida na chamada de função do **dsmInitEx**. Isso permite que usuários não-root acessem arquivos que o root possui se um acesso configurado tiver sido executado.

Utilize a opção *asnodename* na cadeia de opções **dsmInitEx** com a função apropriada para backup, archive, restauração, recuperação, consulta ou exclusão no nome do nó de destino no servidor IBM Spectrum Protect. Consulte "Fazendo Backup de Vários Nós com Suporte ao Proxy de Nó Cliente" na página 83 para obter informações sobre como ativar essa opção.

Gerenciando Espaços no Arquivo

Como os espaços no arquivo são importantes para a operação do sistema, um conjunto separado de chamadas é usado para registrar, atualizar e excluir identificadores de espaço no arquivo. Antes de poder armazenar os objetos associados a um espaço no arquivo, você deve primeiro registrar o espaço no arquivo com o IBM Spectrum Protect.

Utilize a chamada de **dsmRegisterFS** para realizar essa tarefa. Para obter mais informações sobre nomes e IDs do objeto, consulte "IDs e Nomes de Objetos" na página 23.

O identificador do espaço no arquivo é o qualificador de nível superior em uma hierarquia de nome de três partes. O agrupamento de dados relacionados em um espaço no arquivo torna o gerenciamento desses dados mais fácil. Por exemplo, o cliente aplicativo ou o administrador do servidor IBM Spectrum Protect pode excluir um espaço no arquivo e todos os objetos nesse espaço no arquivo.

Os espaços no arquivo também permitem que o aplicativo cliente forneça informações sobre o espaço no arquivo para o servidor e que o administrador pode, então, consultar. Estas informações são retornadas na consulta na estrutura **qryRespFSData** e incluem as seguintes informações do sistema de arquivos:

Tipo	Definição
fstype	O tipo de espaço no arquivo. Esse campo é uma cadeia de caracteres que o cliente aplicativo configura.
fsAttr[platform].fsInfo	Um campo de informações do cliente que é usado para dados específicos do cliente.
capacity	A quantidade total de espaço no espaço no arquivo.

Tipo	Definição
occupancy	A quantidade de espaço que está atualmente ocupado no espaço no arquivo.
backStartDate	O registro de data e hora de quando o último backup foi iniciado (configurado através do envio de uma chamada de **dsmUpdateFS**).
backCompleteDate	O registro de data e hora de quando o último backup foi concluído (configurado através do envio de uma chamada de **dsmUpdateFS**).

A utilização de capacity e occupancy depende do aplicativo cliente. Alguns aplicativos podem não precisar de informações sobre o tamanho do espaço no arquivo, neste caso estes campos podem ter o padrão `0`. Para obter mais informações sobre consultar espaços no arquivo, consulte "Consultando o sistema IBM Spectrum Protect" na página 34.

Após um espaço no arquivo ser registrado com o sistema, é possível fazer backup ou arquivar objetos a qualquer momento. Para atualizar os campos de ocupação e capacidade do espaço no arquivo depois da operação de backup ou archive, chame **dsmUpdateFS**. Esta chamada garante que os valores para a ocupação e capacidade do sistema de arquivos sejam atuais. Você também pode atualizar os campos **fsinfo**, **backupstart** e **backupcomplete**.

Se desejar monitorar as datas de seu último backup, insira uma chamada **dsmUpdateFS** antes de iniciar o backup. Configure a ação de atualização para DSM_FSUPD_BACKSTARTDATE. Isso forçará o servidor a configurar o campo **backStartDate** do espaço no arquivo para a hora atual. Após a conclusão do backup para o espaço no arquivo, digite uma chamada de **dsmUpdateFS** com a ação de atualização que é configurada para DSM_FSUPD_BACKCOMPLETEDATE. Esta chamada cria um registro de data e hora no término do backup.

Se um espaço no arquivo não for mais necessário, ele pode ser excluído pelo comando **dsmDeleteFS**. Em um sistema operacional UNIX ou Linux somente o usuário raiz ou usuários autorizados podem excluir espaços no arquivo.

Os exemplos em Figura 5 na página 28 demonstram como utilizar as três chamadas de espaço no arquivo para UNIX ou Linux. Para obter um exemplo de como usar as três chamadas de espaço no arquivo para Windows, consulte o código do programa de amostra que está instalado em seu sistema.

```
/*  Registre o espaço no arquivo, caso isso ainda não tenha sido feito.  */

dsInt16               rc;
regFSData    fsData;
char         fsName[DSM_MAX_FSNAME_LENGTH];
char         smpAPI[] = "Sample-API";

strcpy(fsName,"/home/tallan/text");
memset(&fsData,0x00,sizeof(fsData));
fsData.stVersion = regFSDataVersion;
fsData.fsName = fsName;
fsData.fsType = smpAPI;
strcpy(fsData.fsAttr.unixFSAttr.fsInfo,"Sample API FS Info");
fsData.fsAttr.unixFSAttr.fsInfoLength =
       strlen(fsData.fsAttr.unixFSAttr.fsInfo) + 1;
fsData.occupancy.hi=0;
fsData.occupancy.lo=100;
fsData.capacity.hi=0;
fsData.capacity.lo=300;

rc = dsmRegisterFS(dsmHandle,fsData);
if (rc == DSM_RC_FS_ALREADY_REGED) rc = DSM_RC_OK;  /* já pronto */
if (rc)
{
   printf("Filespace registration failed: ");
   rcApiOut(dsmHandle, rc);
   free(bkup_buff);
   return (RC_SESSION_FAILED);
}
```

Figura 5. Um Exemplo para Trabalhar com Espaços no Arquivo, Parte 1

```
/* Atualize o espaço no arquivo. */

dsmFSUpd     updFilespace;         /* para atualizar o FS */

updFilespace.stVersion = dsmFSUpdVersion;
updFilespace.fsType = 0;             /* nenhuma alteração */
updFilespace.occupancy.hi = 0;
updFilespace.occupancy.lo = 50;
updFilespace.capacity.hi = 0;
updFilespace.capacity.lo = 200;
strcpy(updFilespace.fsAttr.unixFSAttr.fsInfo,
       "My update for filespace") ;
updFilespace.fsAttr.unixFSAttr.fsInfoLength =
       strlen(updFilespace.fsAttr.unixFSAttr.fsInfo);

updAction = DSM_FSUPD_FSINFO |
            DSM_FSUPD_OCCUPANCY |
            DSM_FSUPD_CAPACITY;

rc = dsmUpdateFS (handle,fsName,&updFilespace,updAction);
printf("dsmUpdateFS rc=%d\n", rc);
```

Figura 6. Um Exemplo para Trabalhar com Espaços no Arquivo, Parte 2

```
/*  Exclua o espaço no arquivo.  */

printf("\nExcluindo o espaço no arquivo
rc = dsmDeleteFS (dsmHandle,fsName,DSM_REPOS_ALL);
if (rc)
{
   printf("  FAILED!!! ");
   rcApiOut(dsmHandle, rc);
}
else printf("  OK!\n");
```

Figura 7. Um Exemplo para Trabalhar com Espaços no Arquivo, Parte 3

Associando Objetos com Classes de Gerenciamento

O recurso principal do IBM Spectrum Protect é a utilização da política (classes de gerenciamento) para definir como objetos são armazenados e gerenciados no armazenamento do IBM Spectrum Protect. Um objeto é associado a uma classe de gerenciamento quando o objeto tem backup ou archive.

Essa classe de gerenciamento determina:
- Quantas versões do objeto são mantidas se for feito backup
- Quanto tempo manter cópias de archive
- Onde inserir o objeto na hierarquia de armazenamento no servidor

As classes de gerenciamento consistem em grupos de cópia de backup e grupos de cópia de archive. Um grupo de cópias é um conjunto de atributos que definem as políticas de gerenciamento para um objeto do qual está sendo feito backup ou archive. Se uma operação de backup estiver sendo executada, os atributos do grupo de cópias de backup se aplicam. Se uma operação de archive estiver sendo executada, os atributos do grupo de cópias de archive se aplicam.

O grupo de cópias de backup ou archive de uma classe de gerenciamento específica pode estar vazio ou `NULL`. Se um objeto estiver ligado a um grupo de cópias de backup `NULL`, não será possível fazer backup desse objeto. Se um objeto for ligado ao grupo de cópias de archive `NULL`, o objeto não poderá ser arquivado.

Como o uso de uma política é um componente muito importante do IBM Spectrum Protect, a API requer que todos os objetos enviados ao servidor tenham primeiramente uma classe de gerenciamento designada, utilizando a chamada de **dsmBindMC**. Com o software do IBM Spectrum Protect, é possível usar uma lista de inclusão-exclusão para afetar a ligação da classe de gerenciamento. O diretório do chamada de **dsmBindMC** utiliza a lista de Inclusão-Exclusão atual para executar a ligação da classe de gerenciamento.

As instruções de inclusão podem associar uma classe de gerenciamento específica a um objeto de backup ou archive. As instruções de exclusão podem evitar que seja feito backup de objetos, mas não que sejam arquivados.

A API requer que **dsmBindMC** seja chamada antes do backup ou archive de um objeto. A chamada de **dsmBindMC** retorna uma estrutura mcBindKey que contém informações sobre a classe de gerenciamento e os grupos de cópias associados ao objeto. Verifique o destino do grupo de cópias antes de prosseguir com um envio. Ao enviar vários objetos em uma única transação, eles devem ter o mesmo destino de grupo de cópias. A chamada de função **dsmBindMC** retorna as seguintes informações:

Tabela 9. Informações Retornadas na Chamada de dsmBindMC

Informações	Descrição
Classe de Gerenciamento	O nome da classe de gerenciamento que foi ligada ao objeto. O cliente aplicativo pode enviar a chamada de **dsmBeginQuery** para determinar todos os atributos dessa classe de gerenciamento.
Grupo de Cópias de Backup	Informa se existe um grupo de cópias de backup para a classe de gerenciamento. Se houver uma operação de backup em execução e não houver um grupo de cópias de backup, esse objeto não poderá ser enviado para armazenamento. Você receberá um código de erro se tiver tentado enviá-lo utilizando a chamada de **dsmSendObj**.
Destino da Cópia de Backup	Esse campo identifica o conjunto de armazenamentos para o qual os dados são enviados. Se você estiver executando uma transação de backup de vários objetos, todos os destinos de cópia dessa transação devem ser o mesmo. Se um objeto tiver um destino de cópia diferente dos objetos anteriores na transação, encerre a transação atual e inicie uma nova transação antes de poder enviar o objeto. Você receberá um código de erro se tentar enviar objetos a destinos de cópias diferentes na mesma transação.
Grupo de Cópias de Archive	Informa se existe um grupo de cópias de archive para a classe de gerenciamento. Se houver uma operação de archive em execução e não houver um grupo de cópias de archive, esse objeto não poderá ser enviado para armazenamento. Você receberá um código de erro se tiver tentado enviá-lo utilizando a chamada de **dsmSendObj**.
Destino da Cópia de Archive	Esse campo identifica o conjunto de armazenamentos para o qual os dados são enviados. Se você estiver executando uma transação de archive de vários objetos, todos os destinos de cópia dessa transação devem ser o mesmo. Se um objeto tiver um destino de cópia diferente dos objetos anteriores na transação, encerre a transação atual e inicie uma nova transação antes de enviar o objeto. Você receberá um código de erro se tentar enviar objetos a destinos de cópias diferentes na mesma transação.

As cópias de backup de um objeto podem ser religadas a uma classe de gerenciamento diferente se um backup subseqüente com o mesmo nome do objeto for feito utilizando uma classe de gerenciamento diferente do original. Por exemplo, se você fizer backup de ObjectA e ligá-lo a Mgmtclass1 e, posteriormente, fizer backup de ObjectA e ligá-lo a Mgmtclass2, o backup mais atual religará as cópias inativas a Mgmtclass2. Os parâmetros definidos em Mgmtclass2 controlariam então todas as cópias. No entanto, os dados não serão movidos se o destino for diferente.

Você também pode religar cópias de backup a uma classe de gerenciamento diferente utilizando a chamada **dsmUpdateObj** ou **dsmUpdateObjEx** com a ação DSM_BACKUPD_MC.

Referências relacionadas:

Opção Deduplication

Classes de Gerenciamento da Consulta

Os aplicativos podem consultar classes de gerenciamento para determinar que as classes de gerenciamento são possíveis para um nó específico e para determinar quais atributos estão na classe de gerenciamento.

Os objetos podem ser ligados às classes de gerenciamento somente através da chamada de **dsmBindMC**. Você pode querer que seu aplicativos consultem os atributos da classe de gerenciamento e exibam os mesmos para os usuários finais. Consulte “Consultando o sistema IBM Spectrum Protect” na página 34 para obter mais informações.

No exemplo na Figura 8, uma instrução de comutação é utilizada para distinguir entre as operações de backup e archive ao chamar **dsmBindMC**. As informações retornadas dessa chamada são armazenadas na estrutura **MCBindKey**.

```
dsUint16_t     send_type;
dsUint32_t     dsmHandle;
dsmObjName   objName;     /* estrutura contendo o nome do objeto   */
mcBindKey    MCBindKey;   /* informações da classe de gerenciamento*/
char         *dest;       /* valor do destino de salvamento        */

switch (send_type)
{
   case (Backup_Send) :
      rc = dsmBindMC(dsmHandle,&objName,stBackup,&MCBindKey);
      dest = MCBindKey.backup_copy_dest;
      break;
   case (Archive_Send) :
      rc = dsmBindMC(dsmHandle,&objName,stArchive,&MCBindKey);
      dest = MCBindKey.archive_copy_dest;
      break;
   default : ;
}

if (rc)
{
   printf("*** dsmBindMC failed: ");
   rcApiOut(dsmHandle, rc);
   rc = (RC_SESSION_FAILED);
   return;
}
```

Figura 8. Um Exemplo de Associação de uma Classe de Gerenciamento a um Objeto

Suspensão e Liberação de Expiração/Exclusão

É possível suspender a exclusão e expiração de objetos de archive específicos, em resposta a uma ação pendente ou em andamento que requer que determinados dados sejam suspensos. No caso de uma ação que possa requerer acesso a dados ser iniciada, esses dados devem estar disponíveis até a ação ser concluída e o acesso aos dados não serem mais requeridos como parte desse processo. Após determinar que a suspensão não é mais necessária (liberada), a sincronização normal de exclusão e expiração continua durante o período de retenção original.

Verifique se o servidor está licenciado, emitindo uma chamada de teste **dsmRetentionEvent**:

1. Consulte um objeto que deseja suspender e obter o ID.
2. Emita **dsmBeginTxn**, **dsmRetentionEvent** com `Hold` e **dsmEndTxn**.
3. Se o servidor não estiver licenciado, você receberá um voto de interrupção com o código de razão `DSM_RC_ABORT_LICENSE_VIOLATION`.

Restrições:

1. Não é possível emitir mais de uma chamada de **dsmRetentionEvent** em uma única transação.
2. Não é possível emitir uma suspensão em um objeto que já está em suspensão.

1. Para suspender objetos, conclua as seguintes etapas:
   a. Consulte o servidor para todos os objetos que deseja colocar em suspensão. Obtenha o ID do objeto para cada objeto.
   b. Emita uma chamada **dsmBeginTxn**, depois emita uma chamada **dsmRetentionEvent** com a lista de objetos, seguida por uma chamada

dsmEventType: eventHoldObj. Se o número de objetos exceder o valor de maxObjPerTxn, utilize várias transações.

c. Use a chamada de função qryRespArchiveData na chamada de função **dsmGetNextQObj** para confirmar se os objetos foram colocados em suspensão. Verifique o valor de objHeld em qryRespArchiveData.

2. Para liberar objetos da suspensão, conclua as seguintes etapas:
   a. Consulte o servidor para todos os objetos que deseja liberar da suspensão. Obtenha o ID do objeto para cada objeto.
   b. Emita uma chamada **dsmBeginTxn**, depois emita uma chamada **dsmRetentionEvent** com a lista de objetos, seguida por uma chamada **dsmEventType**: eventReleaseObj. Se o número de objetos exceder o valor de maxObjPerTxn, utilize várias transações.
   c. Use a resposta **qryRespArchiveData** na chamada de função dsmGetNextQObj para confirmar se os objetos foram liberados da suspensão. Verifique o valor de objHeld em **qryRespArchiveData**.

Proteção de Retenção de Dados de Archive

Os dados controlados pelo IBM Spectrum Protect não podem ser modificados por agentes não autorizados, como um indivíduo ou um programa. Essa proteção é ampliada para evitar a exclusão de dados, como objetos de archive, por qualquer agente antes da expiração do período de retenção.

A proteção da retenção do archive ajuda a assegurar que nenhum indivíduo ou programa possa excluir de forma maliciosa ou acidental os dados controlados pelo IBM Spectrum Protect. Um objeto de archive que é enviado para um servidor de proteção de retenção do archive é protegido contra exclusões acidentais e tem um período de retenção aplicado. A proteção de retenção do archive possui as seguintes restrições:

- Somente operações de archive são permitidas em um servidor de proteção de retenção.
- Qualquer objeto que não é ligado explicitamente a uma classe de gerenciamento através de um valor na chamada de função **dsmBindMc** ou de instruções de inclusão e exclusão será ligado ao nome explícito da classe de gerenciamento padrão. Por exemplo, se a classe de gerenciamento padrão na política do nó for MC1, o objeto será ligado explicitamente a MC1 em vez de a DEFAULT. Em uma resposta de consulta, o objeto seria exibido como ligado a MC1.
- Após ativar a proteção de retenção de dados do archive, qualquer tentativa de excluir um objeto antes de o período de retenção expirar retornará o código DSM_RC_ABORT_DELETE_NOT_ALLOWED na transação final.

Consulte a documentação do servidor IBM Spectrum Protect para obter instruções para a configuração da proteção de retenção para um objeto de archive.

Para configurar a proteção de retenção de dados do archive, conclua as seguintes etapas:

1. Em uma nova instalação do servidor sem dados anteriores, execute o comando **SET ARCHIVERETENTIONPROTECTION ON**.
2. Na sequência de opções da API nas chamadas de função **dsmInit** ou **dsmInitEx**, insira a seguinte instrução:

```
-ENABLEARCHIVERETENTIONPROTECTION=yes
```

Também é possível configurar a opção `enablearchiveretentionprotection` opção em seu arquivo `dsm.opt` em sistemas diferentes de UNIX ou em seu arquivo `dsm.sys` em sistemas UNIX:

```
SERVERNAME srvr1.ret
   TCPPORT                          1500
   TCPSERVERADDRESS                 node.domain.company.com
   COMMMETHOD                       TCPIP
   ENABLEARCHIVERETENTIONPROTECTION YES
```

Para obter informações adicionais sobre esta opção, consulte "A Opção enablearchiveretentionprotection".

3. Emita uma consulta para o servidor para confirmar se o servidor IBM Spectrum Protect está ativado para proteção de retenção do archive. Clique no valor do campo `archiveRetentionProtection` na estrutura `dsmQuerySessInfo`.

A Opção enablearchiveretentionprotection

A opção `enablearchiveretentionprotection` especifica se a proteção de retenção de dados deve ser ativada para objetos de archive no servidor IBM Spectrum Protect dedicado a essa finalidade. O administrador do servidor deve ativar a proteção de retenção de dados em um novo servidor que ainda não tem objetos armazenados (backup, archive ou gerenciado por espaço). Se o aplicativo da API tentar armazenar uma versão de backup ou um objeto gerenciado por espaço no servidor, uma mensagem de erro é emitida.

A nota em Capítulo 3, "Considerações para Projetar um Aplicativo", na página 11 avisa: "Não armazene valores objectID para serem usados para restaurações futuras. Não há garantias de que eles sejam persistentes durante a existência do objeto." pode ser relaxada para aplicativos do gerenciador de archive, já que o servidor do gerenciador de archive não suporta exportação nem importação. Os aplicativos do gerenciador de archive podem salvar e utilizar o ID do objeto para aprimorar o desempenho durante a restauração do objeto.

Se o servidor emitir o comando **`SET ARCHIVERETENTIONPROTECTION ON`**, não será possível excluir um objeto arquivado do servidor utilizando o comando **`delete filespace`** até que os parâmetros de política do grupo de cópias de archive sejam satisfeitos. Consulte a documentação do servidor adequada para obter informações sobre como configurar uma classe de gerenciamento.

Política de Retenção Baseada em Eventos

Em uma política de retenção baseada em evento, o tempo de retenção de um objeto de archive é iniciado por um evento de negócios, como o fechamento de uma conta bancária. A retenção baseada em evento alinha perfeitamente a política de retenção de dados do IBM Spectrum Protect com os requisitos de negócios de dados. Quando o evento ocorrer, o aplicativo enviará um evento **`eventRetentionActivate`** para esse objeto no servidor para iniciar a retenção.

Para usar uma política de retenção baseada em evento, conclua as seguintes etapas:

1. No servidor, crie uma classe de gerenciamento com um archive **`copygroup`** do tipo `EVENT`. Para obter mais informações, consulte a documentação do servidor IBM Spectrum Protect.
2. Consulte a classe de gerenciamento para confirmar se a classe é baseada em evento. Se a classe de gerenciamento for baseada em evento, o campo **`retainInit`** na estrutura **`archDetailCG`** será `ARCH_RETINIT_EVENT`.

3. Ligue os objetos à classe de gerenciamento baseada em evento usando include, **archmc** ou explicitamente através do atributo **mcNameP** na estrutura **ObjAttr** na chamada de função **dsmSendObj**.
4. No ponto em que desejar iniciar a retenção para o objeto, consulte o servidor para todos os objetos afetados. Verifique se eles estão em um estado PENDING e obtenha o ID do objeto. Em um estado pendente, o campo **retentionInitiated** na estrutura **qryRespArchiveData** indica DSM_ARCH_RETINIT_PENDING.
5. Emita uma chamada **dsmBeginTxn**, depois emita uma chamada **dsmRetentionEvent** com a lista de objetos, seguida por uma chamada **dsmEventType**:eventRetentionActivate. Se o número de objetos exceder o valor de maxObjPerTxn, utilize várias transações.

   Restrição: É possível emitir apenas uma chamada **dsmRetentionEvent** por transação.
6. Consulte os objetos para confirmar se a retenção está ativada. Se a retenção estiver iniciada, o campo **retentionInitiated** na estrutura **qryRespArchiveData** terá um valor de I.

Consultando o sistema IBM Spectrum Protect

A API tem várias consultas, como consulta da classe de gerenciamento, que os aplicativos podem utilizar.

Todas as consultas que usam a chamada de **dsmBeginQuery** seguem estas etapas:

1. Envie a chamada de **dsmBeginQuery** com o tipo de consulta apropriado:
   - Cópia de Segurança
   - Arquivamento
   - Objetos com backup ativos
   - Espaço de arquivos
   - Classe de Gerenciamento

   A chamada **dsmBeginQuery** informa a API sobre o formato de dados que é retornado do servidor. Os campos apropriados podem ser colocados nas estruturas de dados que são transmitidas pelas chamadas de **dsmGetNextQObj**. A chamada de consulta inicial também permite que o cliente aplicativo configure o escopo da consulta especificando apropriadamente os parâmetros na chamada de consulta inicial.

   Restrição: Em sistemas UNIX ou Linux, apenas o usuário root pode consultar objetos de backup ativos. Este tipo de consulta é conhecido como "atalho".
2. Digite a chamada de **dsmGetNextQObj** para obter cada registro da consulta. Essa chamada transmite um buffer grande o suficiente para conter dados retornados da consulta. Cada tipo de consulta tem uma estrutura de dados correspondente para os dados retornados. Por exemplo, um tipo de consulta de backup tem uma estrutura **qryRespBackupData** associada que é preenchida quando a chamada **dsmGetNextQObj** é enviada.
3. A chamada de **dsmGetNextQObj** geralmente retorna um dos seguintes códigos:
   - DSM_RC_MORE_DATA: Envie a chamada **dsmGetNextQObj** novamente.
   - DSM_RC_FINISHED: Não há mais dados. Envie a chamada de **dsmEndQuery**.
4. Envie a chamada de **dsmEndQuery**. Quando todos os dados de consulta forem recuperados ou dados de consulta adicionais não forem necessários, insira a

chamada **dsmEndQuery** para terminar o processo de consulta. A API limpa os dados restantes do fluxo de consulta e libera os recursos que foram usados para a consulta.

A Figura 9 exibe o diagrama de estado para operações de consulta.

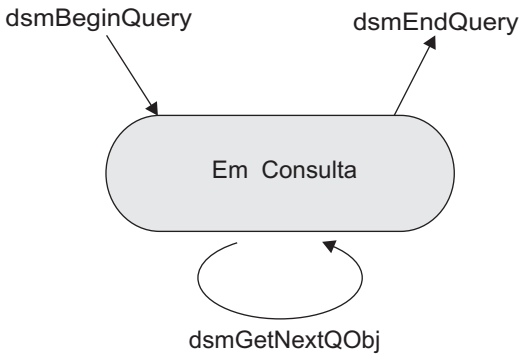


Figura 9. Diagrama de Estado para Consultas Gerais

A Figura 10 exibe o fluxograma para operações de consulta.

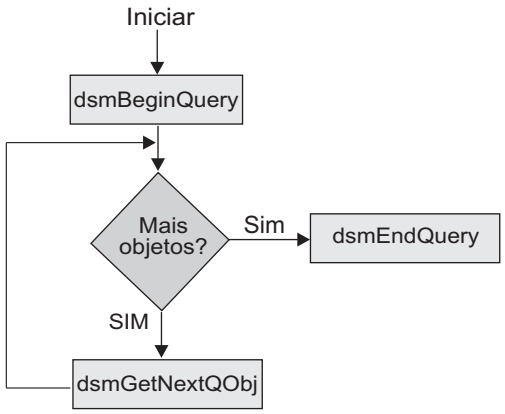


Figura 10. Fluxograma para Consultas Gerais

Exemplo de Consulta ao Sistema

Neste exemplo, uma consulta da classe de gerenciamento imprimi os valores de todos os campos nos grupos de cópias de backup e archive para uma classe de gerenciamento específica.

```
dsInt16                rc;
qryMCData             qMCData;
DataBlk               qData;
qryRespMCDetailData   qRespMCData, *mcResp;
char                  *mc, *s;
dsBool_t                done = bFalse;
dsUint32_t                qry_item;

/* Preencha a estrutura qMCData com os critérios de consulta desejados */
qMCData.stVersion = qryMCDataVersion;   /* versão da estrutura   */
qMCData.mcName    = mc;                 /* nome da classe de gerenciamento */
qMCData.mcDetail  = bTrue;              /* deseja detalhes completos?    */

/* Configure parâmetros do bloco de dados utilizado para obter e enviar dados */
qData.stVersion = DataBlkVersion;
qData.bufferLen = sizeof(qryRespMCDetailData);
qData.bufferPtr = (char *)&qRespMCData;

qRespMCData.stVersion = qryRespMCDetailDataVersion;
```

```
if ((rc = dsmBeginQuery(dsmHandle,qtMC,(dsmQueryBuff *)&qMCData)))
{
   printf("*** dsmBeginQuery failed: ");
   rcApiOut(dsmHandle, rc);
   rc = (RC_SESSION_FAILED);
}
else
{
   done = bFalse;
   qry_item = 0;
   while ( !done )
   {
      rc = dsmGetNextQObj(dsmHandle,&qData);
      if ((   (rc == DSM_RC_MORE_DATA)
           || (rc == DSM_RC_FINISHED))
          && qData.numBytes)
      {
         qry_item++;
         mcResp = (qryRespMCDetailData *)qData.bufferPtr;
         printf("Mgmt.          printf("              Name:
         printf("   Backup CG Name:
            .
            .  /* outros campos dos grupos de cópias de backup e archive */
            .
         printf("   Copy Destination:
      }
      else
      {
         done = bTrue;
         if (rc != DSM_RC_FINISHED)
         {
            printf("*** dsmGetNextQObj failed: ");
            rcApiOut(dsmHandle, rc);
         }
      }
      if (rc == DSM_RC_FINISHED) done = bTrue;
   }
   rc = dsmendQuery (dsmHandle);
}
```

Figura 11. Um Exemplo de Execução de uma Consulta no Sistema

Eficiência do Servidor

Use essas diretrizes quando recuperar ou enviar objetos para o servidor IBM Spectrum Protect.

- Ao recuperar objetos do servidor IBM Spectrum Protect, siga estas diretrizes:
  - Recupere os dados na ordem de restauração fornecida pelo servidor IBM Spectrum Protect. A ordem de restauração é especialmente importante para

dispositivos de fita, pois a recuperação de dados que não for ordenada poderá resultar em rebobinamentos de fita e montagens.
  - Mesmo que os dados sejam armazenados em um dispositivo de disco, você poderá economizar tempo quando as recuperações forem ordenadas.
  - Execute o máximo de trabalho possível em uma única sessão do servidor IBM Spectrum Protect.
  - Não inicie e pare várias sessões.
- Ao enviar objetos para o servidor IBM Spectrum Protectsiga estas diretrizes:
  - Envie vários objetos em uma única transação.
  - Evite enviar um objeto por transação, especialmente quando os dados forem enviados diretamente para um dispositivo de fita. Parte da transação do dispositivo de fita é garantir que os dados nos buffers de RAM da fita sejam gravados em mídia.

Conceitos relacionados:

"Selecionando e Classificando Objetos pela Ordem de Restauração" na página 69

Informações relacionadas:

"Iniciando ou Encerrando uma Sessão" na página 16

Enviando Dados a um Servidor

A API permite que os clientes aplicativos enviem dados ou objetos denominados e seus dados associados ao armazenamento do servidor IBM Spectrum Protect.

Dica: Você pode fazer backup ou arquivar dados. Execute todas as operações de envio de uma transação.

O Modelo da Transação

O envio de todos os dados para o armazenamento do IBM Spectrum Protect durante uma operação de backup ou archive ocorre em uma transação. Um modelo de transação fornece um alto nível de integridade de dados, mas impõe algumas restrições que devem ser levadas em consideração pelo aplicativo cliente.

Inicie uma transação por uma chamada a **dsmBeginTxn** ou encerre uma transação por uma chamada a **dsmEndTxn**. Uma única transação é uma ação atômica. Os dados enviados dentro dos limites de uma transação são confirmados no sistema no final da transação ou recuperados se a transação for encerrada prematuramente.

As transações podem consistir em envios de um único objeto ou em envios de vários objetos. Para melhorar o desempenho do sistema diminuindo a carga do sistema, envie objetos menores em uma transação de vários objetos. O cliente aplicativo determina se transações simples ou múltiplas são apropriadas.

Envie todos os objetos em uma transação de vários objetos para o mesmo destino de cópia. Se precisar enviar um objeto para um destino diferente do do objeto anterior, encerre a transação atual e inicie uma nova. Na nova transação, é possível enviar o objeto para o novo destino de cópia.

Nota: Os objetos que não contêm nenhum dado de bit (*sizeEstimate=0*) não são verificados para consistência do destino de cópia.

O IBM Spectrum Protect limita o número de objetos que podem ser enviados em uma transação com vários objetos. Para saber esse limite, chame **dsmQuerySessInfo**

e examine o campo **maxObjPerTxn**. Esse campo exibe o valor da opção *TXNGroupmax* que está configurada em seu servidor.

O cliente aplicativo deve monitorar os objetos enviados em uma transação para executar o processo de nova tentativa ou o processo de erro se a transação for encerrada prematuramente. O servidor ou o cliente pode parar uma transação a qualquer momento. O cliente aplicativo deve estar preparado para manipular encerramentos de transação inesperados que não iniciou.

Agregação de Arquivo

Os servidores do IBM Spectrum Protect utiliza, uma função chamada de agregação de arquivo. Com a agregação de arquivo, todos os objetos enviados em uma única transação são armazenados juntos, o que economiza espaço e aprimora o desempenho. Você ainda pode consultar e restaurar os objetos separadamente.

Para utilizar essa função, todos os objetos de uma transação devem ter o mesmo nome de espaço no arquivo. Se o nome do espaço no arquivo for alterado em uma transação, o servidor fecha o objeto agregado existente e inicia um novo.

Transferência de dados sem LAN

A API pode tirar proveito de transferência de dados sem a LAN se a opção **dsmSetUp** para multiencadeamento estiver ON. A API retorna a existência de um destino sem a LAN na resposta da **Classe de Gerenciamento de Consulta**, estrutura de resposta **archDetailCG** ou **backupDetailCG**, campo **bLanFreeDest**.

Você pode utilizar operações sem a LAN em plataformas que são suportadas pelo agente de armazenamento. A plataforma Macintosh está excluída.

As informações sem a LAN são fornecidas nas estruturas de saída a seguir. A estrutura de saída (**dsmEndGetDataExOut_t**) para **dsmEndGetData** inclui o campo **totalLFBytesRecv**. Esse é o número total de bytes sem a LAN que são recebidos. A estrutura de saída (**dsmEndSendObjExOut_t**) para **dsmEndSendObjEx** inclui o campo **totalLFBytesSent**. Esse é o número total de bytes sem a LAN que foram enviados.

Informações relacionadas:

Movimentação de dados sem a LAN: visão geral do agente e armazenamento

Operações de gravação simultânea

É possível configurar os conjuntos de armazenamento do servidor IBM Spectrum Protect para gravar simultaneamente em um conjunto de armazenamento primário e um conjunto ou conjuntos de armazenamento de cópia durante um backup ou archive. Use esta configuração para criar diversas cópias do objeto.

Se uma operação de gravação simultânea falhar, o código de retorno na função **dsmEndTxn** poderá ser DSM_RC_ABORT_STGPOOL_COPY_CONT_NO, o que indica que a gravação em um dos conjuntos de armazenamento de cópia falhou e a opção de conjunto de armazenamento do IBM Spectrum Protect COPYCONTINUE foi configurada para NO. O aplicativo é encerrado e o problema deve ser resolvido pelo administrador do servidor do IBM Spectrum Protect.

Para obter mais informações sobre configurar operações de gravação simultânea, consulte a documentação do servidor IBM Spectrum Protect.

Aprimorando o Desempenho da API

Você pode usar as opções do cliente `tcpbuffsize` e `tcpnodelay` e o parâmetro da API **`DataBlk`** para aperfeiçoar o desempenho da API.

A Tabela 10 descreve as ações que podem ser tomadas para aprimorar o desempenho da API.

Tabela 10. Opções de Backup-Archive e o Parâmetro da API que Aperfeiçoam o Desempenho

Opções do Cliente de Backup-Archive	**Descrição**
`tcpbuffsize`	Especifica o tamanho do buffer do TCP. O valor padrão é 31 KB. Para aprimorar o desempenho, configure o valor para 32 KB.
`tcpnodelay`	Especifica se pequenos buffers devem ser enviados ao servidor em vez de mantê-los. Para aprimorar o desempenho, configure esta opção para *`yes`* para todas as plataformas. Esta opção é válida apenas para Windows e AIX.
Parâmetro da API	**Descrição**
`DataBlk`	Este parâmetro é usado com a chamada de função **`dsmSendData`** para determinar o tamanho do buffer do aplicativo. Para obter melhores resultados, configure o parâmetro como múltiplo do valor *`tcpbuffsize`* que é especificado com o `tcpbuffsize` menos 4 bytes. Por exemplo, configure um valor de 28 para este parâmetro quando o valor de `tcpbuffsize` for configurado para 32 KB.

Cada chamada **`dsmSendData`** é síncrona e não retorna até que os dados transferidos para a API no **`dataBlkPtr`** sejam limpos para a rede. A API inclui uma sobrecarga de 4 byte para cada buffer da transação que é posicionado na rede.

Por exemplo, quando o tamanho do buffer da transação for 32 KB e tamanho do buffer do **`DataBlk`** do aplicativo for 31 KB, então cada bufer do **`DataBlk`** do aplicativo se ajusta em um buffer de comunicações e não pode ser limpo imediatamente. Entretanto, se o buffer do **`DataBlk`** do aplicativo for exatamente 32 KB, e por causa da API estar incluindo 4 bytes por buffer de transação, duas limpezas são necessárias: uma de 32 KB e uma de 4 bytes. Além disso, se você configurar a opção `tcpnodelay` para `no`, limpar os 4 bytes pode levar até 200 milissegundos.

Configurar a API para Enviar Dados de Desempenho ao Monitor de Desempenho do Cliente

O monitor de desempenho do cliente é um componente do Tivoli Storage Manager Administration Center usado para exibir os dados de desempenho coletados pela API. O monitor de desempenho do cliente registra e exibe os dados de desempenho para operações de backup de cliente, archive e restauração.

Com o monitor de desempenho ativado, é possível exibir dados de desempenho que são coletados pela API usando o monitor de desempenho; o monitor de desempenho está disponível no Tivoli Storage Manager Administration Center. A partir da versão 7.1, o componente Administration Center não estará mais incluído nas distribuições do Tivoli Storage Manager ou do IBM Spectrum Protect. Se você tiver um Administration Center que foi instalado com uma liberação do servidor anterior, será possível continuar usando-o para exibir dados de desempenho. Se você ainda não tiver um Administration Center instalado, será possível fazer

download da versão liberada anteriormente a partir de ftp://public.dhe.ibm.com/storage/tivoli-storage-management/maintenance/admincenter/v6r3/. Para obter informações sobre o uso do monitor de desempenho, consulte a documentação do servidor Tivoli Storage Manager Versão 6.3.

Configurando as Opções do Monitor de Desempenho do Cliente

Você permite que clientes IBM Spectrum Protect usem o monitor de desempenho, especificando parâmetros no arquivo de opções do cliente. Especifique estas opções para cada cliente que você deseja monitorar.

Ao monitorar o desempenho em computadores UNIX e Linux, configure o limite do descritor de arquivo aberto para pelo menos 1024, usando o seguinte comando:

```
ulimit -n 1024
```

Para configurar as opções do monitor de desempenho do cliente, conclua as seguintes etapas:

1. Abra o arquivo de opções do cliente para cada cliente que está sendo monitorado. Dependendo de sua configuração, as opções do cliente estarão em um dos seguintes arquivos:
   - `dsm.opt`
   - `dsm.sys`
2. Inclua as seguintes opções no arquivo de opções do cliente:

   ```
   PERFMONTCPSERVERADDRESS
   PERFMONTCPPORT
   PERFMONCOMMTIMEOUT
   ```

PERFMONTCPSERVERADDRESS

A opção PERFMONTCPSERVERADDRESS especifica o nome do host ou endereço IP do sistema onde o monitor de desempenho do cliente está instalado.

Clientes Suportados

Esta opção é independente de plataforma e é suportada por todos os clientes.

Arquivo de opções

Configure esta opção no arquivo de opções do cliente (`dsm.opt` ou `dsm.sys`).

Sintaxe

Parâmetros

server
O nome do host do servidor ou endereço IP do sistema em que o monitor de desempenho do cliente está instalado (é o mesmo servidor que executa o Centro de Administração).

Exemplos

Arquivo de opções:

```
PERFMONTCPSERVERADDRESS 131.222.10.5
```

Linha de comandos:

Esta opção não pode ser configurada usando a linha de comandos.

PERFMONTCPPORT

O número da porta em que o monitor de desempenho do cliente atende para dados de desempenho dos clientes.

Clientes Suportados

Esta opção é independente de plataforma e é suportada por todos os clientes.

Arquivo de opções

Configure esta opção no arquivo de opções do cliente (`dsm.opt` ou `dsm.sys`).

Sintaxe

Parâmetros

port

A porta que é monitorada para dados de desempenho do cliente. A porta 5129 é a porta padrão.

Exemplos

Arquivo de opções:

```
PERFMONTCPPPORT 5000
```

Linha de comandos:

Esta opção não pode ser configurada usando a linha de comandos.

PERFMONCOMMTIMEOUT

Especifica o tempo máximo, em segundos, que a chamada dsmTerminate aguarda pela chegada dos dados de desempenho após uma sessão ser finalizada.

Clientes Suportados

Esta opção é independente de plataforma e é suportada por todos os clientes.

Arquivo de opções

Configure esta opção no arquivo de opções do cliente (`dsm.opt` ou `dsm.sys`).

Sintaxe

Parâmetros

seconds
O tempo para aguardar pela chegada do restante dos dados de desempenho, antes de terminar a sessão.

Exemplos

Arquivo de opções:

```
PERFMONCOMMTIMEOUT 60
```

Linha de comandos:
Esta opção não pode ser configurada usando a linha de comandos.

Enviando Objetos ao Servidor

Os clientes aplicativos podem enviar dados ou objetos denominados e seus dados associados para o armazenamento do IBM Spectrum Protect utilizando as funções de backup e archive da API. Os componentes de backup e archive do sistema permitem o uso de diferentes procedimentos de gerenciamento para os dados que são enviados para o armazenamento.

O atributo de estimativa de tamanho é uma estimativa do tamanho total do objeto de dados a enviar ao servidor. Se o aplicativo não conhece o tamanho exato do objeto, configure *sizeEstimate* para uma estimativa mais alta. Se a estimativa for menor do que o tamanho real, o servidor IBM Spectrum Protect utilizaria recursos extras para gerenciar alocações de espaço extra.

Dicas:

- Seja o mais preciso possível quando fizer esta estimativa de tamanho. O servidor usa esse atributo para a alocação de espaço e o posicionamento de objetos eficientes em seus recursos de armazenamento.
- Se a estimativa for menor do que o tamanho real, um servidor com armazenamento em cache não alocará espaço adicional e interromperá o envio.

Você pode encontrar problemas se *sizeEstimate* for muito grande. O servidor pode não ter espaço suficiente para o tamanho estimado, mas tem espaço para o tamanho real; ou o servidor pode usar dispositivos mais lentos.

Você pode fazer backup ou arquivar objetos com mais de dois gigabytes de tamanho. Os objetos podem ser compactados ou descompactados.

Para iniciar uma operação de envio, chame **dsmSendObj**. Se você tiver mais dados do que pode enviar de uma vez, poderá fazer chamadas repetidas a **dsmSendData** para transferir o restante das informações. Chame **dsmEndSendObj** para concluir a operação de envio.

Entendendo Objetos de Backup e Archive

O componente de backup do sistema IBM Spectrum Protect suporta várias versões de objetos denominados que são armazenados no servidor.

Qualquer objeto com backup no servidor que tenha o mesmo nome que um objeto que já está armazenado no servidor a partir do cliente está sujeito a controle de versão. Os objetos são considerados como nos estados ativo ou inativo no servidor. A cópia mais recente de um objeto no servidor que não foi desativada está no estado ativo. Qualquer outro objeto com o mesmo nome, seja uma versão mais antiga ou uma cópia desativada, é considerado inativo. As construções de classe de gerenciamento definem critérios de gerenciamento diferentes. Elas são designadas a objetos ativos e inativos no servidor.

Tabela 11 lista os campos do grupo de cópias que se aplicam aos estados ativo e inativo:

Tabela 11. Campos de Grupos de Cópias de Backup

Campo	Descrição
VEREXISTS	O número de versões inativas se existirem versões ativas.
VERDELETED	O número de versões inativas se versões ativas não existirem.
RETEXTRA	O número de dias para manter versões inativas.
RETONLY	O número e dias para manter as últimas versões inativas se as versões ativas não existirem.

Se as versões de backup tiverem um nome exclusivo, como utilizar um registro de data e hora no nome, então, a definição de versão não ocorrerá automaticamente: todos os objetos estarão ativos. Os objetos ativos nunca expiram, portanto, um aplicativo seria responsável por desativá-los com a chamada de **dsmDeleteObj**. Nessa situação, o aplicativo precisaria que os objetos desativados expirassem assim que possível. O usuário definiria um grupo de cópias de backup com VERDELETED=0 e RETONLY=0.

O componente de archive do sistema IBM Spectrum Protect permite que objetos sejam armazenados no servidor com controles de períodos de retenção ou expiração em vez de controle de versão. Cada objeto armazenado é exclusivo, apesar de seu nome poder ser igual ao de um objeto já arquivado. Os objetos de archive têm um campo de descrição associado com os metadados que podem ser utilizados durante a consulta para identificar um objeto específico.

Cada objeto no servidor IBM Spectrum Protect recebe um ID de objeto exclusivo. A persistência do valor original não é garantido durante a vida de um objeto (especificamente após uma exportação ou importação). Portanto, um aplicativo não deve consultar e salvar o ID do objeto original para utilização em restaurações posteriores. Em vez disso, um aplicativo deve salvar o nome do objeto e a data de inserção. Essas informações podem ser utilizadas durante uma restauração para consultar objetos e verificar a data de inserção. Em seguida, o ID do objeto atual pode ser utilizado para restaurar o objeto.

Compactação

Opções de configuração em um dado nó e a opção **dsmSendObj** objCompressed determinam se o IBM Spectrum Protect compacta o objeto durante o envio. Além disso, os objetos com sizeEstimate inferior a DSM_MIN_COMPRESS_SIZE nunca serão compactados.

Se o objeto já estiver compactado (objCompressed=bTrue), ele não é compactado novamente. Se ele não estiver compactado, o IBM Spectrum Protect decidirá se o objeto será compactado, com base nos valores da opção de compactação configurada pelo administrador e que é configurada nas origens de configuração da API.

O administrador pode alterar os limites de compactação no servidor, utilizando o comando de nó de registro (compression=yes, no ou client-determined). Se for client-determined, o comportamento de compactação será determinado pelo valor da opção de compactação nas origens de configuração.

Alguns tipos de dados, como os dados que já estão compactados, podem na verdade ser maiores quando processados com o algoritmo de compactação. Quando isso ocorre, o código de retorno DSM_RC_COMPRESS_GREW é gerado. Se você perceber que isso pode ocorrer, mas deseja que a operação de envio continue mesmo assim, informe aos usuários finais para especificar a seguinte opção no arquivo de opções:

```
COMPRESSAlways Yes
```

Se, durante uma função **dsmSendData**, com compactação ativada, você obtiver um código de retorno DSM_RC_COMPRESS_GREW, você pode desejar iniciar do princípio e enviar o objeto novamente sem compactação. Para reforçar isto, configure **dsmSendObj** ObjAttr.objCompressed para bTrue.

As informações sobre o comportamento de compactação real durante um **dsmSendObj** são retornadas pela chamada de **dsmEndSendObjEx**. objCompressed especifica se a compactação foi realizada. totalBytesSent é o número de bytes enviados pelo aplicativo. totalCompressedSize é o número de bytes após a compactação. A chamada de **dsmEndSendObjEx** também tem um campo totalLFBytesSent que contém o total de bytes enviados sem a LAN.

Atenção: Se seu aplicativo planeja utilizar restauração ou recuperação parcial do objeto, não é possível compactar os dados durante o envio. Para reforçar isto, configure **dsmSendObj** ObjAttr.objCompressed para bTrue.

Tipo de compactação

O tipo de compactação utilizado pelo cliente é determinado pela combinação de compactação e deduplicação de dados do lado do cliente utilizada durante o processo de backup ou de archive.

O algoritmo de compactação utilizado pelo cliente é relatado pela API em um novo campo nas estruturas **qryRespArchiveData** e **qryRespBackupData**:

```
dsChar_t        compressAlg[20];  /* compression algorithm name */
```

São relatados os seguintes tipos de compactação:

LZ4 Um método de compactação mais rápido e eficiente que é usado pelo cliente quando um objeto deduplicado pelo cliente é enviado para um conjunto de armazenamentos de contêiner compatível com LZ4 no servidor IBM Spectrum Protect. O servidor deve estar na versão 7.1.5 ou posterior, e

deve usar conjuntos de armazenamentos de contêiner. A compactação LZ4 do lado do cliente é usada somente quando a deduplicação de dados do lado do cliente está ativada.

LZW Um tipo tradicional de compactação que pode ser usado pelo cliente em qualquer uma das seguintes situações:

- Objetos deduplicados pelo cliente são enviados para conjuntos de armazenamentos tradicionais (não de contêiner) no servidor.
- O objeto do cliente não passa pela deduplicação de dados do lado do cliente.
- O objeto do cliente passa somente pela deduplicação de dados do lado do servidor tradicional.

Campo em branco
O objeto não é compactado pelo cliente. O objeto não é compactado porque a opção `compression` está configurada como *no* ou a opção não foi especificada durante o processamento de backup ou de archive. Embora o objeto não seja compactado pelo cliente, ele pode ser compactado pelo servidor.

O tipo de compactação não é configurável. Ele é determinado pelo cliente de backup-archive no momento do processo de backup ou de archive.

Exemplo

O exemplo a seguir mostra o campo `Tipo de compactação` na saída das consultas de backup e archive a partir do aplicativo de amostra de 64 bits **`dapismp`**:

```
Enter selection ==>1
                       Filespace:\fs1
                       Highlevel:\hl
                        Lowlevel:\ll
             Object Type(D/F/A):f
   Active(A),Inactive(I),Both(B):a
If root, query all owners? (Y/N):
              Object Owner Name:
   point in time date (MMDDYYYY):
   point in time time     (hhmm):
     Show detailed output? (Y/N):y
On Restore, Wait for mount?(Y/N):
Are the above responses correct (y/n/q)?


Item 1: \fs1\hl\ll
   Object type: File
   Object state: Active
   Insert date: 2016/2/3 10:57:41
   Expiration date: 0/0/0 0:0:0
   Proprietário:
   Restore order: 0-0-0-0-0
   Object id: 0-40967
   Copy group: 1
   Media class: Fixed
   Mgmt class: DEFAULT
   Object info is        :IBM Spectrum Protect API Verify Data
   Object info length is :73
   Estimated size : 0  4000
   Compression : YES
   Compression Type: LZ4
   Encryption : NO
   Encryption Strength : NONE
   Client Deduplicated : YES
```

Eliminação da Cópia do Buffer

A função de eliminação da cópia de buffer remove a cópia dos buffers de dados entre um aplicativo e o servidor IBM Spectrum Protect, o que resulta em melhor utilização do processador. Para efeito máximo, use esta abordagem em um ambiente sem a LAN.

Os buffers para movimentação de dados são alocados pelo IBM Spectrum Protect e um ponteiro é passado de volta para o aplicativo. O aplicativo posiciona os dados no buffer fornecido e este buffer é passado através das camadas de comunicação para o agente de armazenamento usando memória compartilhada. Os dados são, então, movidos para o dispositivo de fita, que elimina cópias de dados. Essa função pode ser utilizada com as operações de backup ou archive.

Atenção: Quando usar este método, preste atenção extra para obter manipulação de buffer adequada e tamanhos de buffer. Os buffers são compartilhados entre os componentes e qualquer memória sobrescrita que seja um resultante de um erro de programação resultará em erros graves.

A seqüência geral de chamadas para backup/archive é a seguinte:

```
dsmInitEx (UseTsmBuffers = True, numTsmBuffers = [a quantidade de IBM Spectrum Protect que
           o aplicativo precisa alocar)
dsmBeginTxn
for each object in the txn
  dsmBindMC
    dsmSendObject
      dsmRequestBuffer
      dsmSendBufferData (envia e libera o buffer usado)
    dsmEndSendObjEx
dsmEndTxn
for each buffer still held
      dsmReleaseBuffer
dsmTerminate
```

A função **dsmRequestBuffer** pode ser chamada diversas vezes, até o valor que é especificado pela a opção `numTsmBuffers`. Um aplicativo pode ter dois encadeamentos: um encadeamento do produtor que preenche buffers com dados e um encadeamento do consumidor que envia esses buffers para IBM Spectrum Protect com a chamada **dsmSendBufferData**. Quando uma chamada **dsmRequestBuffer** é emitida e o **numTsmBuffers** é atingido, a chamada **dsmRequestBuffer** bloqueia até que um buffer seja liberado. A liberação do buffer pode acontecer chamando **dsmSendBufferData**, que envia e libera um buffer, ou chamando **dsmReleaseBuffer**. Para obter mais informações, consulte `callbuff.c` no diretório de amostra do API.

Se em qualquer ponto houver uma falha no envio, o aplicativo deve liberar todos os buffers que estão retidos e terminar a sessão. Por exemplo:

```
If failure
  for each data buffer held by application
    call dsmReleaseBuffer
dsmTerminate
```

Se um aplicativo chamar **dsmTerminate** e um buffer estiver retido, o API não existe. O código a seguir é retornado: `DSM_RC_CANNOT_EXIT_MUST_RELEASE_BUFFER`. Se o aplicativo não puder liberar o buffer, ele deve sair do processo para forçar a limpeza.

Eliminação, Restauração e Recuperação de Cópias de Buffer

O servidor IBM Spectrum Protect controla a quantia de dados a ser posicionada no buffer, com base na otimização de acesso à fita com restauração e recuperação. Este método não é como um benefício para o aplicativo como o método normal de obter dados. Durante a criação do protótipo, verifique o desempenho do método de eliminação da cópia de buffer e use este método apenas se vir uma melhoria válida.

A quantidade máxima de dados em um único buffer retornados pelo servidor IBM Spectrum Protect é (256K bytes – sobrecarga do cabeçalho). Como uma consequência, apenas aplicativos que lidam com gravações de buffer pequeno se beneficiam deste mecanismo de recuperação dos dados. O aplicativo deve dar atenção especial ao número de bytes em buffer, dependendo do tamanho do objeto, na rede, e outras condições de limite. Em algumas situações, a utilização da eliminação da cópia do buffer pode na verdade ter um desempenho pior do que a restauração normal. A API normalmente armazena em cache dados e retorna uma duração fixa do aplicativo. O aplicativo pode, então, controlar o número de gravações de dados de volta para o disco.

Se você usar eliminação de cópia de buffer, crie um mecanismo de armazenamento em cache de dados para buffers que são menos que o tamanho do buffer de gravação preferencial. Por exemplo, se um aplicativo grava blocos de dados de 64K no disco, o aplicativo deve executar estas ações:

1. Chamar **dsmGetBufferData**.
2. Gravar blocos de 64K.
3. No bloco final, copiar o restante em um **tempBuff**, emitir outra chamada **dsmGetBufferData** e preencher **tempBuff** com o resto dos dados.
4. Continuar a gravar blocos de 64K:

```
dsmGetBufferData #1 get 226K    dsmGetBufferData #2 get 240K
Block1 64K - gravação em disco   Block1 30K - cópia para tempbuff - gravação em disco
Block2 64K - gravação em disco   Block2 64K - gravação em disco
Block3 64K - gravação em disco   Block3 64K - gravação em disco
Block4 34K - cópia para tempbuff  Block4 64K - gravação em disco
Block5 18K - gravação em tempbuff  etc
```

Neste exemplo, seis gravações de disco são direcionadas e 1 está em cache.

A sequência geral de chamadas para restaurar e recuperar é como a seguir:

dsmInitEx (`UseTsmBuffers` = `True` `numTsmBuffers` = quantos buffers o aplicativo deseja alocar).

```
dsmBeginGetData
While obj id
  dsmGetObj (nenhum dado restaurado nesse buffer de chamada configurado para NULL)
 While data to read
   dsmGetBufferData (returns the data in the data buffer)
  ...process data...
  dsmReleaseBuffer
dsmEndGetObj
dsmEndGetData
```

Para toda chamada **dsmGetBufferData**, implemente uma chamada **dsmReleaseBuffer**. **dsmGetBufferData** e **dsmReleaseBuffer** correspondente não precisam ser consecutivas. Um aplicativo pode emitir várias chamadas de **dsmGetBufferData** primeiro para obter vários buffers e, em seguida, emitir as chamadas de**dsmReleaseBuffer** correspondentes. Para obter o código de amostra que usa esta função, consulte `callbuff.c` no diretório de amostra da API.

Restrição: Como a API fornece o buffer e o objetivo é minimizar a utilização do processador, não é permitido o processamento adicional de dados no buffer. O aplicativo não pode usar criptografia e comprimir com eliminação de cópia de buffer porque ambas as operações requerem processamento e cópias de dados.

Implemente o caminho de movimentação de dados regular e a eliminação de cópia de buffer para permitir que o usuário alterne entre ambos os caminhos com base em suas necessidades. Se o usuário deve compactar ou criptografar dados, então use o mecanismo existente. Se houver uma restrição de processador, use o novo mecanismo. Ambos os mecanismos são complementares e não substituem completamente o outro.

Criptografia da API

Dois métodos estão disponíveis para criptografar dados: criptografia gerenciada por aplicativo e criptografia do cliente IBM Spectrum Protect.

Selecione e use apenas um desses métodos para criptografar dados. Os métodos são mutuamente exclusivos e, se você criptografar dados usando ambos os métodos, não será possível restaurar ou recuperar alguns dados. Por exemplo, presuma que um aplicativo use a criptografia gerenciada por aplicativos para criptografar o objeto A e, em seguida, use a criptografia de cliente do IBM Spectrum Protect para criptografar o objeto B. Durante uma operação de restauração, se o aplicativo configurar a opção para usar a criptografia do cliente do IBM Spectrum Protect e ela tentar restaurar ambos os objetos, apenas o objeto B poderá ser restaurado; o objeto A não poderá ser restaurado porque foi criptografado pelo aplicativo, não pelo cliente.

Independentemente do método de criptografia usado, o IBM Spectrum Protect deve ativar a autenticação de senha. Por padrão, o servidor usa `SET AUTHENTICATION ON`.

A API usa a criptografia AES de 128 bits ou AES de 256 bits. A criptografia de dados AES de 256 bits fornece um nível mais alto de criptografia de dados do que a criptografia de dados AES de 128 bits. Os arquivos cujo backup foi feito usando a criptografia AES de 256 bits não podem ser restaurados com um cliente anterior. A criptografia pode ser ativada com ou sem compactação. Se usar a criptografia, você não poderá usar as funções de restauração e recuperação de objetos parciais e de eliminação da cópia de buffer.

Criptografia gerenciada por aplicativo

Com a criptografia gerenciada por aplicativo, o aplicativo fornece a senha principal para a API (usando a chave DSM_ENCRYPT_USER) e é responsabilidade do aplicativo gerenciar a senha principal.

Atenção: Se a chave de criptografia não for salva e você tiver esquecido a chave, seus dados se tornarão irrecuperáveis.

O aplicativo fornece a senha da chave na chamada de **dsmInitEx** e deve fornecer a senha da chave apropriada na hora da restauração.

Atenção: Se a senha principal for perdida, não haverá como restaurar os dados.

A mesma senha de chave deve ser usada para operações de backup e restauração (ou archive e recuperação) para o mesmo objeto. Esse método não depende do nível do servidor IBM Spectrum Protect. Para configurar esse método, o aplicativo precisa seguir estas etapas:

1. Configure a variável bEncryptKeyEnabled para bTrue na chamada para **dsmInitEx** e configure a variável encryptionPasswordP para apontar para uma cadeia com a senha da chave de criptografia.
2. Configure include.encrypt para os objetos a serem criptografados. Por exemplo, para criptografar todos os dados, configure:

   ```
   include.encrypt /.../* (UNIX)
   ```

   e

   ```
   include.encrypt *\...\* (Windows)
   ```

   Para criptografar o objeto /FS1/DB2/FULL, configure:

   ```
   include.encrypt /FS1/DB2/FULL
   ```
3. Configure ENCRYPTKEY=PROMPT|SAVE na sequência de opções que é transmitida para a API na chamada **dsmInitEx** no Windows. Esta opção também pode ser configurada em dsm.opt (Windows) ou dsm.sys (UNIX ou Linux).

Por padrão, a opção encryptkey é configurada para prompt. Essa configuração assegura que a chave não é armazenada automaticamente. Se encryptkey save for especificada, a chave será armazenada pelo IBM Spectrum Protect na máquina local, mas somente uma chave poderá ser válida para todas as operações do IBM Spectrum Protect com o mesmo nome de nó.

Depois do envio de um objeto, **dsmEndSendObjEx** especifica se um objeto foi criptografado e qual método foi usado. Valores possíveis no campo *encryptionType*:

- DSM_ENCRYPT_NO
- DSM_ENCRYPT_USER
- DSM_ENCRYPT_CLIENTENCRKEY

A tabela a seguir lista os tipos de criptografia de API, pré-requisitos e funções que estão disponíveis.

Tabela 12. Tipos de Criptografia, Pré-requisitos e Funções Disponíveis da API

Tipo	Pré-requisito	Função disponível
ENCRYPTIONTYPE	Nenhum	Configure ENCRYPTIONTYPE na sequência de opções que é transmitida para a API na chamada **dsmInitEx** no Windows. ENCRYPTIONTYPE=AES128 por padrão.
EncryptKey=save	Nenhum	API e backup/archive
EncryptKey=prompt	Nenhum	API e backup/archive
EncryptKey=generate	Nenhum	API e backup/archive
EnableClientEncryptKey	Nenhum	Somente para a API

Nota: Recomenda-se que o servidor esteja com a autenticação ATIVADA. Se a autenticação for OFF, a chave não será criptografada, mas os dados ainda serão criptografados. No entanto, isso não é recomendado.

A Tabela 13 na página 50 mostra como os Usuários Autorizados e os Usuários Não Autorizados podem criptografar ou decriptografar dados durante uma operação de backup ou restauração, dependendo do valor que é especificado para a opção passwordaccess. O arquivo TSM.PWD deve existir para executar as operações a seguir de usuário autorizado e usuário não-autorizado. O usuário autorizado cria o arquivo TSM.PWD e configura a opção encryptkey para salvar e a opção passwordaccess para gerar.

Tabela 13. Criptografando ou Decriptografando Dados com Chave Gerenciada pelo Aplicativo no UNIX ou no Linux

Operação	Opção `passwordaccess`	Opção `encryptkey`	Resultado
Backup de usuário autorizado	generate	save	Dados criptografados.
	generate	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	save	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
Restauração de usuário autorizado	generate	save	Dados criptografados.
	generate	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	save	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
Backup do usuário não autorizado	generate	save	Dados criptografados.
	generate	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	save	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
Restauração de usuário não autorizado	generate	save	Dados criptografados.
	generate	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	save	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.
	prompt	prompt	Os dados são criptografados se `encryptionPasswordP` contiver uma senha de criptografia.

Criptografia do Cliente IBM Spectrum Protect

A criptografia de cliente do IBM Spectrum Protect usa a chave que é gerenciada pelo valor DSM_ENCRYPT_CLIENTENCRKEY para proteger seus dados. A criptografia do cliente é transparente para o aplicativo que está usando a API, com exceção de que operações de restauração e operações de recuperação de objetos parciais não são possíveis para objetos que foram criptografados ou compactados.

Para a criptografia de cliente do IBM Spectrum Protect e para a criptografia gerenciada pelo aplicativo, a senha de criptografia refere-se a um valor de sequência que é usado para gerar a chave de criptografia real. O valor para a

opção de senha de criptografia é de 1-63 caracteres de comprimento, mas a chave que é gerada a partir dele é sempre de 8 bytes para 56 DES, 16 bytes para 128 AES e 32 bytes para 256 AES.

Atenção: Se a chave de criptografia não estiver disponível, os dados não poderão ser restaurados ou recuperados. Ao usar `ENABLECLIENTENCRYPTKEY` para criptografia, a chave de criptografia é armazenada no banco de dados do servidor. Para objetos que usam este método, o banco de dados do servidor deve existir e ter os valores apropriados para os objetos para uma restauração apropriada. Certifique-se de que você faça backup do banco de dados do servidor frequentemente para evitar perda de dados.

Esse é o método mais simples para implementar, no qual uma chave de criptografia aleatória é gerada por sessão e armazenada no servidor IBM Spectrum Protect com o objeto no banco de dados do servidor. Durante a restauração, a chave armazenada é utilizada para decriptografia. Usando esse método, o gerenciamento da chave é a responsabilidade do IBM Spectrum Protect e o aplicativo simplesmente não precisa lidar com a chave Como a chave é armazenada no banco de dados do servidor, deve-se ter um banco de dados válido do IBM Spectrum Protect para uma operação de restauração de um objeto criptografado. Quando a chave é transmitida entre a API e o servidor, ela também é criptografada. A transmissão da chave é segura e, quando a chave está armazenada no banco de dados do servidor do IBM Spectrum Protect, ela é criptografada. A única vez que a chave aparece sem criptografia com o fluxo de dados de exportação é quando os dados de um nó são exportados entre servidores.

Para ativar a criptografia do cliente do IBM Spectrum Protect, conclua as seguintes etapas:

1. Especifique `-ENABLECLIENTENCRYPTKEY=YES` na sequência de opções que é transmitida à API na chamada dsmInitEx ou configure a opção no arquivo de opções do sistema `dsm.opt` (Windows) ou `dsm.sys` (UNIX ou Linux).
2. Configure `include.encrypt` para os objetos a serem criptografados. Por exemplo, para criptografar todos os dados, configure:

   ```
   include.encrypt /.../* (UNIX)
   ```

   e

   ```
   include.encrypt *\...\* (Windows)
   ```

   Para criptografar o objeto `/FS1/DB2/FULL`, configure:

   ```
   include.encrypt /FS1/DB2/FULL
   ```

Deduplicação de dados

A deduplicação de dados é um método de reduzir as necessidades de armazenamento, eliminando dados redundantes.

Visão Geral

Dois tipos de deduplicação de dados estão disponíveis no IBM Spectrum Protect: *deduplicação de dados do cliente* e *deduplicação de dados do servidor*.

Deduplicação de dados do lado do cliente é uma técnica de deduplicação de dados que é usada no cliente de backup-archive para remover dados redundantes durante o processamento de backup e archive antes de os dados serem transferidos para o servidor do IBM Spectrum Protect. O uso da deduplicação de dados do lado do cliente pode reduzir a quantidade de dados enviada ao cliente por uma rede local.

A *deduplicação de dados do lado do servidor* é uma técnica de deduplicação de dados feita pelo servidor. O administrador do IBM Spectrum Protect pode especificar o local de deduplicação de dados (cliente ou servidor) a ser usado com o parâmetro **DEDUP** no comando do servidor **REGISTER NODE** ou **UPDATE NODE**.

Aprimoramentos

Com a deduplicação de dados do lado do cliente, você pode:

- Excluir arquivos específicos em um cliente a partir da deduplicação de dados.
- Ativar um cache de deduplicação de dados que reduz o tráfego de rede entre o cliente e o servidor. O cache contém extensões que foram enviadas para o servidor nas operações de backup incremental anteriores. Em vez de consultar o servidor quanto à existência de uma extensão, o cliente consulta seu cache.

  Especifique um tamanho e local para um cache cliente. Se uma inconsistência entre o servidor e o cache local for detectada, o cache local será removido e preenchido novamente.

  Nota: Para aplicativos que usam a API do IBM Spectrum Protect, o cache de deduplicação de dados não deve ser usado por causa do potencial de falhas de backup causadas pelo cache que está fora de sincronização com o servidor IBM Spectrum Protect. Se múltiplas sessões de cliente de backup-archive simultâneas estão configuradas, deverá haver um cache separado configurado para cada sessão.
- Ative a deduplicação e a compactação de dados do lado do cliente para reduzir a quantidade de dados que é armazenada pelo servidor. Cada extensão é compactada antes de ser enviada ao servidor. A negociação é feita entre economia de armazenamento e a energia de processamento necessária para compactar dados de cliente. Em geral, se você compactar e deduplicar dados no sistema do cliente, estará usando, aproximadamente, duas vezes mais a energia de processamento que a deduplicação de dados usa sozinha.

  O servidor pode trabalhar com dados deduplicados, compactados. Além disso, clientes de backup-archive anteriores à V6.2 podem restaurar dados deduplicados, compactados.

A deduplicação de dados do lado do cliente usa o seguinte processo:

- O cliente cria extensões. As *extensões* fazem parte dos arquivos que são comparados com outras extensões de arquivos para identificar duplicações.
- O cliente e o servidor trabalham juntos para identificar extensões duplicadas. O cliente envia extensões não-duplicadas para o servidor.
- Operações de deduplicação de dados de cliente subsequentes criam novas extensões. Algumas ou todas essas extensões podem corresponder às extensões criadas nas operações de deduplicação de dados anteriores e enviadas ao servidor. Extensões correspondentes não são enviadas ao servidor novamente.

Benefícios

A deduplicação de dados do lado do cliente oferece várias vantagens:

- Pode reduzir a quantidade de dados enviados através da rede local (LAN).
- A energia de processamento necessária para identificar dados duplicados é transferida dos nós servidor para cliente. A deduplicação de dados do lado do servidor é sempre ativada para conjuntos de armazenamentos ativados para

deduplicação. No entanto, arquivos que estão nos conjuntos de armazenamentos ativados pela deduplicação e que foram deduplicados pelo cliente não exigem processamento adicional.

- A energia de processamento necessária para remover dados duplicados no servidor é eliminada, permitindo que as economias de espaço no servidor ocorram imediatamente.

A deduplicação de dados do lado do cliente tem uma possível desvantagem. O servidor não tem cópias inteiras dos arquivos de cliente *até* que você faça backup dos conjuntos de armazenamentos primários que contêm extensões de cliente para um conjunto de armazenamento de cópia não deduplicado. (As *extensões* são partes de um arquivo e são criadas durante o processo de deduplicação de dados.) Durante o backup do conjunto de armazenamentos para um conjunto de armazenamentos não deduplicados, as extensões de cliente são remontadas em arquivos contíguos.

Por padrão, os conjuntos de armazenamentos de acesso sequencial primários que são configurados para a deduplicação de dados devem ter backup para conjuntos de armazenamentos de cópias não deduplicados, antes de serem recuperados e antes de os dados deduplicados serem removidos. O padrão assegura que o servidor sempre tenha cópias de todos os arquivos, em um conjunto de armazenamentos primários ou em um conjunto de armazenamento de cópia.

Importante: Para obter mais redução de dados, a deduplicação e compactação de dados do lado do cliente poderão ser ativadas em conjunto. Cada extensão é compactada antes de ser enviada ao servidor. A compactação economiza espaço, mas aumenta o tempo de processamento na estação de trabalho do cliente.

As seguintes opções estão relacionadas à deduplicação de dados:

- Deduplicação
- Dedupcachepath
- Dedupcachesize
- Enablededupcache
- Exclude.dedup
- Include.dedup

Deduplicação de Dados do Lado do Cliente pela API

A *deduplicação de dados do lado do cliente* é usada pela API no cliente de backup-archive para remover dados redundantes durante o processamento de backup e archive, antes da transferência dos dados para o servidor IBM Spectrum Protect.

A deduplicação de dados do lado do cliente é usada pela API para remover dados redundantes durante o processamento de backup e archive, antes da transferência dos dados para o servidor IBM Spectrum Protect. O uso da deduplicação de dados do lado do cliente pode reduzir a quantidade de dados enviada ao cliente por uma rede local. O uso da deduplicação de dados do lado do cliente também pode reduzir o espaço de armazenamento do servidor IBM Spectrum Protect.

Quando o cliente é ativado para a deduplicação de dados do lado cliente e você executa backup ou operação de archive, os dados são enviados para o servidor como extensões. Na próxima vez em que o backup ou a operação de archive for

executada, o cliente e o servidor identificarão quais extensões de dados já passaram por backup ou archive e enviarão apenas as extensões de dados exclusivas ao servidor.

Para a deduplicação de dados do lado do cliente, o servidor e a API devem estar na versão 6.2 ou mais recente.

Antes de você usar a deduplicação de dados do lado do cliente para fazer backup ou archive de seus arquivos, o sistema deverá atender aos seguintes requisitos:

- O cliente deve ter a opção `deduplication` ativada.
- O servidor deve ativar o cliente para deduplicação de dados do lado do cliente com o parâmetro **DEDUP=CLIENTORSERVER** no comando **REGISTER NODE** ou **UPDATE NODE**.
- O destino do conjunto de armazenamentos para os dados deve ser um conjunto de armazenamentos ativado para a deduplicação de dados. O conjunto de armazenamentos ativado para deduplicação de dados é um tipo de dispositivo do arquivo somente.
- Verifique se os arquivos estão vinculados à classe de gerenciamento correta.
- Um arquivo pode ser excluído do processamento de deduplicação de dados do lado do cliente. Por padrão, todos os arquivos estão incluídos.
- Os arquivos devem ter mais de 2 KB.
- O servidor pode limitar o tamanho máximo da transação para a deduplicação de dados configurando a opção `CLIENTDEDUPTXNLIMIT` no servidor. Consulte as informações da documentação do servidor sobre essa opção.

Se algum desses requisitos não for atendido, os dados serão processados normalmente, sem deduplicação de dados do lado do cliente.

A seguir estão algumas restrições de deduplicação de dados:

- A movimentação de dados sem LAN e a deduplicação de dados do lado do cliente são mutuamente exclusivas. Se você ativar a movimentação de dados sem LAN e a deduplicação de dados do lado do cliente, as operações de movimentação de dados sem LAN serão concluídas, e a deduplicação de dados do lado do cliente será ignorada.
- A criptografia e a deduplicação de dados do lado do cliente são mutuamente exclusivas. Se você ativar a criptografia e a deduplicação de dados do lado do cliente, as operações de criptografia serão concluídas, e a deduplicação de dados do lado do cliente será ignorada. Os arquivos criptografados e os arquivos elegíveis para deduplicação de dados do lado do cliente podem ser processados na mesma operação, mas isso é feito em transações separadas.

  Requisitos:

  1. Em qualquer transação, todos os arquivos devem ser incluídos para deduplicação de dados ou excluídos. Se tiver arquivos combinados, a transação falhará, e um código de retorno do `DSM_RC_NEEDTO_ENDTXN` será retornado pela API.
  2. Use a criptografia do dispositivo de armazenamento juntamente com a deduplicação de dados do lado do cliente. Como o SSL é usado juntamente com a deduplicação do lado do cliente, não haverá necessidade da criptografia do cliente.
- As funções a seguir não estão disponíveis para a deduplicação de dados do lado do cliente:
  – Cliente do IBM Spectrum Protect for Space Management (HSM)

  - Buffer compartilhado da API
  - NAS
  - Backup de subarquivo
- A eliminação da cópia de buffer não pode ser usada sem transformações de dados, como compactação, criptografia e deduplicação de dados.
- Se você usar a deduplicação no lado do cliente, a API detectará e falhará (com RC=254) os backups de extensões de arquivo que estejam marcados como expirados no servidor durante o envio de dados ao servidor. Se desejar tentar novamente a operação, será necessário incluir essa programação no aplicativo de chamada.
- Operações de gravação simultânea no servidor terão precedência sobre a deduplicação de dados do lado do cliente. Se as operações de gravação simultânea forem ativadas, a deduplicação de dados do lado do cliente não ocorrerá.

  Restrição: Quando a deduplicação de dados do lado do cliente for ativada, a API não poderá ser recuperada de um estado em que o servidor ficou sem armazenamento no conjunto de destino, mesmo que haja um próximo conjunto definido. Um código de razão de parada DSM_RS_ABORT_DESTINATION_POOL_CHANGED é retornado e a operação falha. Há duas maneiras de recuperar desta situação:

  1. Peça ao administrador que inclua mais volumes utilizáveis no conjunto de arquivos original.
  2. Tente a operação novamente com a deduplicação de dados desativada.

Para obter maior economia de largura de banda, você pode ativar um cache local para a deduplicação de dados. O cache local evita a necessidade de consultas no servidor IBM Spectrum Protect. O valor padrão do ENABLEDEDUPCACHE é NO, para que o cache não fique fora de sincronização com o servidor. Se o cache estiver fora de sincronização com o servidor, o aplicativo reenviará todos os dados. Caso o aplicativo possa tentar novamente executar uma transação com falha e você queira usar o cache local, configure a opção ENABLEDEDUPCACHE como YES no arquivo dsm.opt (Windows) ou dsm.sys (UNIX).

No fim de uma restauração, se *todos* os dados forem restaurados por meio da API, e o objeto for deduplicado pelo cliente, uma compilação de ponta a ponta será calculada e comparada ao valor calculado no momento do backup. Se esses valores não se corresponderem, o erro DSM_RC_DIGEST_VALIDATION_ERROR será retornado. Se um aplicativo receber esse erro, os dados serão corrompidos. Esse erro também pode ser um resultado de um erro temporário na rede, portanto, tente a restauração ou a recuperação novamente.

A seguir é exibido um exemplo do comando de sessão de consulta que mostra as informações de deduplicação de dados:

```
dsmQuerySessInfo Values:
Server Information:
Server name: SERVER1
Server Host: AVI
Server port: 1500
Server date: 2009/10/6 20:48:51
Server type: Windows
Server version: 6.2.0.0
Server Archive Retention Protection : NO
Client Information:
Client node type: API Test1
Client filespace delimiter: :
Client hl & ll delimiter: \
```

```
Client compression: Client determined (3u)
Client archive delete: Client can delete archived objects
Client backup delete: Client CANNOT delete backup objects
Maximum objects in multiple object transactions: 4096
Lan free Enabled: NO
Deduplication : Client Or Server
General session info:
Node: AVI
Proprietário:
API Config file:
```

A seguir é exibido um exemplo do comando de classe de gerenciamento de consulta que mostra as informações da duplicação de dados:

```
Policy Information:
Domain name: DEDUP
Policyset name: DEDUP
Policy activation date: 0/0/0 0:0:0
Default management class: DEDUP
Backup retention grace period: 30 days
Archive retention grace period: 365 days
Mgmt. Class 1:
Name: DEDUP
Description: dedup - values like standard
Backup CG Name: STANDARD
Frequency: 0
Ver. Data Exists: 2
Ver. Data Deleted: 1
Retain Extra Ver: 30
Retain Only Ver: 60
Copy Destination: AVIFILEPOOL
Lan free Destination: NO
Deduplicate Data: YES

Archive CG Name: STANDARD
Frequency: 10000
Retain versions: 365
Copy Destination: AVIFILEPOOL
Lan free Destination: NO
Retain Init : CREATE
Retain Minimum : 65534
Deduplicate Data: YES
```

Referências relacionadas:

Opção Deduplication

Excluir Arquivos da Deduplicação de Dados

É possível optar por excluir os arquivos de backup ou archive da deduplicação de dados.

Para excluir arquivos do processamento de deduplicação de dados, siga estas etapas:

1. Configure a opção `exclude.dedup` para os objetos a serem excluídos.

   Por exemplo, para excluir todos os dados deduplicados para sistemas UNIX, configure:

   ```
   exclude.dedup /.../*
   ```

2. Para excluir todos os dados dedup de sistemas Windows, configure:

   ```
   exclude.dedup *\...\*
   ```

Importante: Se um objeto for enviado a um conjunto de deduplicação de dados, a deduplicação ocorrerá no servidor, mesmo que o objeto seja excluído da deduplicação de dados do lado do cliente.

Incluir Arquivos para Deduplicação de Dados

É possível optar por incluir os arquivos de backup ou archive da deduplicação de dados.

Para refinar a lista de arquivos a serem incluídos, a opção `include.dedup` pode ser usada em conjunto com a opção `exclude.dedup`.

Por padrão, todos os objetos elegíveis estão incluídos para deduplicação de dados.

A seguir estão alguns exemplos para UNIX e Linux:

```
exclude.dedup /FS1/.../*

include.dedup /FS1/archive/*
```

A seguir estão alguns exemplos para Windows:

```
exclude.dedup E:\myfiles\...\*

include.dedup E:\myfiles\archive\*
```

Deduplicação de Dados do Lado do Servidor

A deduplicação de dados do lado do servidor é a deduplicação de dados que é executada pelo servidor.

O administrador do IBM Spectrum Protect pode especificar o local de deduplicação de dados (cliente ou servidor) a ser usado com o parâmetro **DEDUP** no comando do servidor **REGISTER NODE** ou **UPDATE NODE**.

Em um conjunto de armazenamentos ativado para deduplicação de dados (conjunto de arquivos), somente uma instância de uma extensão de dados é retida. Outras instâncias das mesmas extensões de dados são substituídas por um ponteiro para a instância retida.

Para obter mais informações sobre a deduplicação de dados do lado do servidor, consulte a documentação de servidor IBM Spectrum Protect.

Failover do Aplicativo

Quando o servidor IBM Spectrum Protect fica indisponível devido a uma interrupção, os aplicativos que usam a API podem executar failover automaticamente em um servidor secundário para recuperação de dados.

O servidor IBM Spectrum Protect ao qual o cliente e a API se conectam durante processos de produção normais é chamado de *servidor principal*. Quando o servidor principal está configurado para replicação de nó, esse servidor também é conhecido como o *servidor de replicação de origem*. Os dados do nó cliente no servidor de replicação de origem podem ser replicados no *servidor de replicação de destino*. Esse servidor também é conhecido como o *servidor secundário*, e é o servidor no qual o cliente executa failover automaticamente quando o servidor principal falha.

O cliente e a API devem ser configurados para failover automatizado de cliente e deve se conectar a um servidor versão 7.1 (ou posterior) que replica dados do nó cliente. A configuração para a API é igual à configuração para o cliente de backup-archive.

Durante operações normais, as informações de conexão do servidor secundário são enviadas automaticamente para o cliente do servidor principal durante o processo de logon. As informações do servidor secundário são salvas automaticamente no arquivo de opções do cliente.

Sempre que o aplicativo cliente efetua logon no servidor IBM Spectrum Protect, ele tenta entrar em contato com o servidor principal. Se o servidor principal estiver indisponível, o aplicativo executará failover automaticamente no servidor secundário, usando as informações do servidor secundário no arquivo de opções do cliente. No modo de failover, o aplicativo pode consultar o servidor secundário e restaurar ou recuperar dados replicados.

Você deve fazer backup do aplicativo pelo menos uma vez no servidor principal. A API pode executar failover no servidor secundário para recuperar dados apenas se os dados do nó cliente tiverem sido replicados do servidor principal para o servidor secundário.

Conceitos relacionados:

Configuração e uso de failover de cliente automatizados

Informações de Status de Failover

A API fornece informações de status que os aplicativos podem utilizar para determinar o status de failover e o status de dados de cliente replicados no servidor secundário.

O status de replicação indica se o backup mais recente foi replicado no servidor secundário. Se o registro de data e hora da operação de backup mais recente na API corresponder ao registro de data e hora do backup no servidor secundário, o status de replicação será atual. Se os dois registros de data e hora não corresponderem, o status de replicação não será atual e os dados replicados poderão estar desatualizados.

As informações de status de replicação a seguir são retornadas na resposta **query filespace** na chamada de função **dsmGetNextQObj** na estrutura **qryRespFSData**:

Tabela 14. Informações de status de replicação relatadas pela API

Informações de status	Tipo	Definição
Início da última replicação	**lastReplStartDate**	A última vez que a replicação foi iniciada.
Final da última replicação	**lastReplCmpltDate**	A última vez que a replicação foi concluída, mesmo que tenha ocorrido uma falha.
Data do último armazenamento de backup (Servidor)	**lastBackOpDateFromServer**	O registro de data e hora do último armazenamento que foi salvo no servidor.
Data do último armazenamento de backup (Local)	**lastBackOpDateFromLocal**	O registro de data e hora do último armazenamento que foi salvo no cliente.

O status de failover é relatado pelo campo **bIsFailOverMode** na estrutura **dsmInitExOut_t**.

Consulte Apêndice B, “Arquivos de Origem de Definições de Tipos de API”, na página 163 para as definições de estrutura e tipo da API.

O código de retorno DSM_RC_SIGNON_FAILOVER_MODE indica que o cliente e a API executaram failover no servidor secundário e estão em execução no modo de failover.

Exemplo de Conexão Durante um Failover

A saída de exemplo a seguir é um exemplo de conexão no servidor durante um failover:

```
signon
Doing signon for node khoyt, owner , with password khoytpass
ANS2106I Connection to primary IBM Spectrum Protect server 123.45.6.78 failed

ANS2107I Attempting to connect to secondary server TARGET at 123.45.6.79 : 1501

ANS2108I Connected to secondary server TARGET.
Handle on return = 1

************************************************************
After dsmInitEx:
Server TARGET ver/rel/lev 7/1/0/0
userNameAuthorities      : Owner
Replication Server name  : TARGET
Home Server name         : MINE
Connected to replication server
************************************************************
```

Exemplo de Comando de Sessão de Consulta

A saída de amostra a seguir é um exemplo do comando **query session** que mostra as informações do servidor secundário (replicação):

```
query session
dsmQuerySessInfo Values:
  Server Information:
    Server name   : TARGET
    Server Host   : 123.45.6.79
    Server port: 1500
    Server date   : 2013/5/21  14:13:32
    Server type: Windows
    Server version: 7.1.0.0
    Server Archive Retention Protection : NO
  Replication Server Infomation
    Home Server name        : MINE
    Replication Server name : TARGET
       Host                 : 123.45.6.79
       Porta: 1501
    Fail over status        : Connected to replication server
  Client Information:
    Client node type          : Unix
    Client filespace delimiter: /
    Client hl & ll delimiter  : /
    Client compression: Client determined (3u)
    Client archive delete: Client can delete archived objects
    Client backup delete: Client CANNOT delete backup objects
    Maximum objects in multiple object transactions: 4096
    Lan free Enabled: NO
    Deduplication             : Server Only
  General session info:
    Node           : KHOYT
    Access Node    :
    Proprietário:
    API Config file:
  Policy Information:
    Domain name                   : STANDARD
    Policyset name                : STANDARD
    Policy activation date: 0/0/0 0:0:0
    Default management class      : STANDARD
    Backup retention grace period: 30 days
    Archive retention grace period: 365 days
```

Exemplo de Comando de Espaço no Arquivo de Consulta

A saída de amostra a seguir é um exemplo do comando **query filespace** que mostra o status de replicação de um espaço no arquivo no servidor secundário:

```
filespace query
Filespace pattern to query:*
Are the above responses correct (y/n/q)?

Filespace Name  Type            Occupancy  Capacity   Start         End
--------------- --------------- ---------- ---------- ------------- -------------
/fs             API:Sample             100        300   0/0/0 0:0:0   0/0/0 0:0:0

     Start of last Replication :  2013/5/21 21:3:2
     End of last Replication   :  2013/5/21 21:3:3
                                  Server                 Local
     Last backup store date    :  2013/5/21 21:18:25     2013/5/21 21:18:25
     Last archive store date   :  0/0/0 0:0:0            0/0/0 0:0:0
     Last HSM store date       :  0/0/0 0:0:0            0/0/0 0:0:0
     FSINFO : Sample API FS Info
```

Referências relacionadas:

"dsmGetNextQObj" na página 112

Diagramas de Fluxo de Exemplo para Backup e Archive

A API foi projetada para fluxos de lógica direta e transições claras entre os vários estados do cliente aplicativo. Essa transição de estado limpo capta as falhas de lógica e erros de programa no início do ciclo de desenvolvimento, aprimorando muito a qualidade e a confiabilidade do sistema.

Por exemplo, não é possível fazer uma chamada de **dsmSendObj** a menos que uma transação tenha sido iniciada e uma chamada de **dsmBindMC** tenha sido feita anteriormente ao objeto do qual está sendo feito backup.

A Figura 12 exibe o diagrama de estado para execução de operações de backup ou archive em uma transação. A seta que aponta de "Em Objeto de Envio" para **dsmEndTxn** indica que uma chamada de **dsmEndTxn** pode ser iniciada após uma chamada de **dsmSendObj** ou **dsmSendData**. Você pode querer fazer isso se tiver ocorrido uma condição de erro durante o envio de um objeto e você deseja parar toda a operação. Nesse caso, você deve utilizar um voto de DSM_VOTE_ABORT. Sob circunstâncias normais, no entanto, chame **dsmEndSendObj** antes de encerrar a transação.

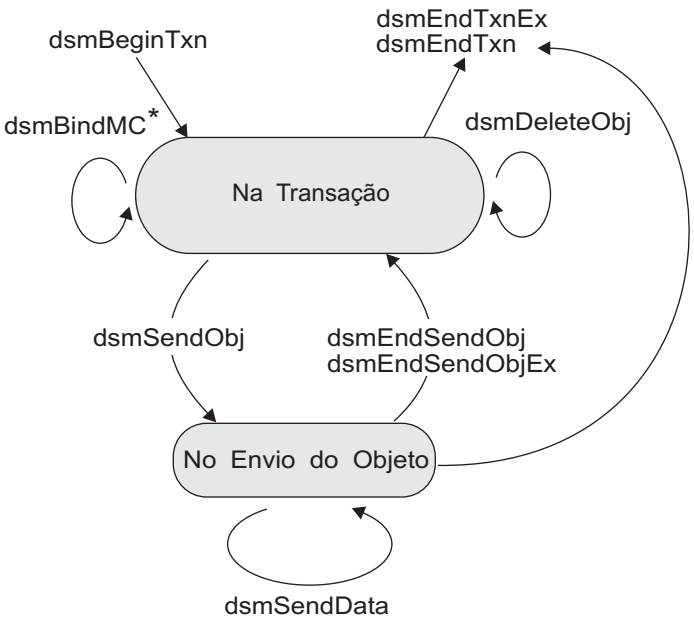


Figura 12. Diagrama de Estado para Operações de Backup e Archive

A Figura 13 na página 62 exibe o fluxograma para execução de operações de backup ou archive em uma transação.

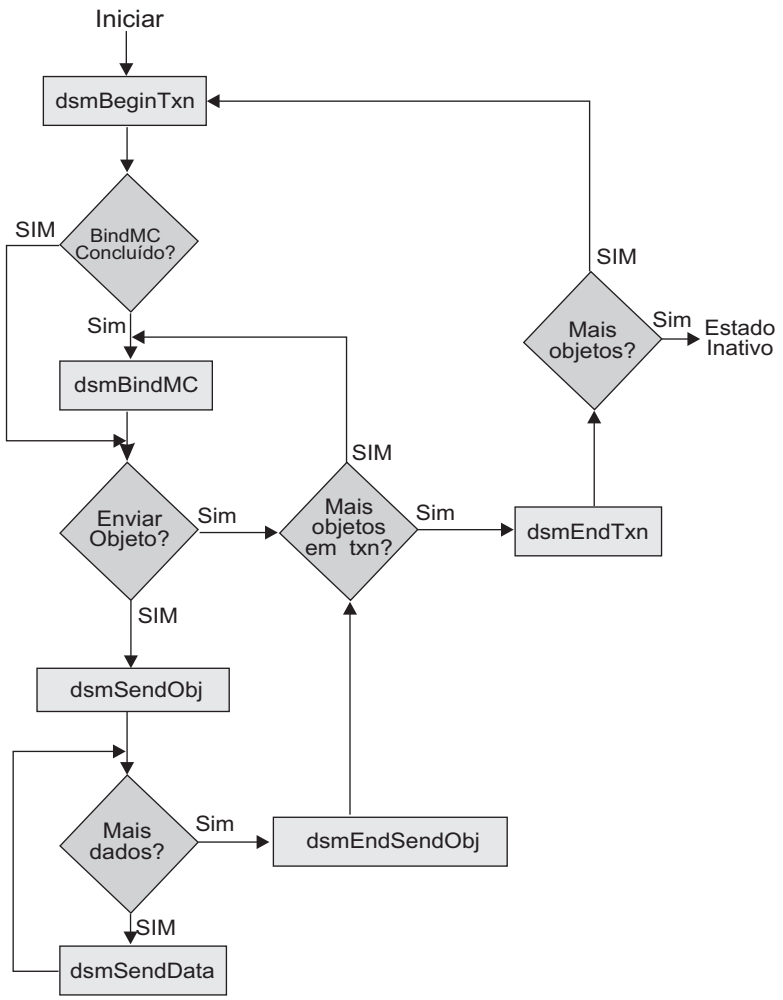


Figura 13. Fluxograma para Operações de Backup e Archive

O recurso principal desses dois diagramas é o loop entre as seguintes chamadas da API a partir de uma transação:

- **dsmBindMC**
- **dsmSendObj**
- **dsmSendData**
- **dsmEndSendObj**

A chamada de **dsmBindMC** é exclusiva, de forma que é possível iniciá-la de dentro ou de fora dos limites de uma transação. Também é possível iniciá-la a partir de uma transação diferente, se necessário. O único requisito para a chamada de **dsmBindMC** é que é feita antes do backup ou archive de um objeto. Se o objeto do qual está fazendo backup ou arquivando não estiver associado a uma classe de gerenciamento, um código de erro é retornado de **dsmSendObj**. Nessa situação, a transação é encerrada chamando **dsmEndTxn** (essa condição de erro não é mostrada no fluxograma).

O fluxograma ilustra como um aplicativo utilizaria várias transações de objeto. Mostra onde os pontos de decisão podem ser colocados para determinar se o objeto enviado se ajusta à transação ou se uma nova transação pode ser iniciada.

Exemplo de código de funções da API que enviam dados para armazenamento do IBM Spectrum Protect

Este exemplo demonstra o uso das funções da API que enviam dados para o armazenamento do IBM Spectrum Protect. A chamada de **dsmSendObj** aparece dentro de uma instrução de comutação, de forma que diferentes parâmetros possam ser chamados, dependendo de se uma operação de backup ou archive está sendo executada.

A chamada de **dsmSendData** é chamada a partir de um loop que envia dados repetidamente até um sinalizador ser configurado, permitindo que a execução do programa saia do loop. Toda a operação de envio é executada a partir da transação.

O terceiro parâmetro na chamada de **dsmSendObj** é um buffer que contém a descrição do archive. Como os objetos de backup não têm uma descrição, esse parâmetro é NULL ao fazer backup de um objeto.

A Figura 8 na página 31 exibe um exemplo que mostra a utilização da chamada de função **dsmBindMC**.

```
if ((rc = dsmBeginTxn(dsmHandle)) )       /* Identificador da sessão da API */
{
   printf("*** dsmBeginTxn failed: ");
   rcApiOut(dsmHandle, rc);
   return;
}

/*  Chame dsmBindMC se ainda não tiver feito isso anteriormente */
objAttr.sizeEstimate.hi = 0;       /* estimativa de      */
objAttr.sizeEstimate.lo = 32000;   /* tamanho do objeto */
switch (send_type)
{
   case (Backup_Send) :
      rc = dsmSendObj(dsmHandle,stBackup,
      NULL,&objName,&objAttr,NULL);
      break;
   case (Archive_Send) :
      archData.stVersion = sndArchiveDataVersion;
      archData.descr = desc;
      rc = dsmSendObj(dsmHandle,stArchive,
   &archData,&objName,&objAttr,NULL);
      break;
   default : ;
}
if (rc)
{
   printf("*** dsmSendObj failed: ");
   rcApiOut(dsmHandle, rc);
   return;
}
done = bFalse;
while ( !done )
{
   dataBlk.stVersion = DataBlkVersion;
   dataBlk.bufferLen = send_amt;
   dataBlk.numBytes = 0;
   dataBlk.bufferPtr = bkup_buff;
   rc = dsmSendData(dsmHandle,&dataBlk);
   if (rc)
   {
      printf("*** dsmSendData failed: ");
      rcApiOut(dsmHandle, rc);
      done = bTrue;
   }
   /* Ajuste o buffer de dataBlk para a próxima parte a enviar */
}
rc = dsmEndSendObj(dsmHandle);
if (rc)
{
   printf("*** dsmEndSendObj failed: ");
   rcApiOut(dsmHandle, rc);
}
txn_reason = 0;
rc = dsmEndTxn(dsmHandle,           /* identificador da sessão da API */
            DSM_VOTE_COMMIT,      /* Confirme a transação         */
            &txn_reason);         /* Razão se txn for abortado*/
if (rc || txn_reason)
{
   printf("*** dsmEndTxn failed: rc = ");
   rcApiOut(dsmHandle, rc);
   printf("   reason =
}
```

Figura 14. Um Exemplo de Envio de Dados a um Servidor

Agrupamento de Arquivos

A API do IBM Spectrum Protect tem um protocolo de agrupamento de arquivo lógico que relata diversos objetos individuais juntos. É possível fazer referência e gerenciar esses grupos como um grupo lógico no servidor. Um grupo lógico requer que todos os membros do grupo e o líder do grupo pertençam ao mesmo nó e espaço no arquivo no servidor.

Cada grupo lógico tem um líder de grupo. Se o líder do grupo for excluído, o grupo será excluído. Não é possível excluir um membro que faz parte de um grupo. A expiração de todos os membros de um grupo depende do líder do grupo. Por exemplo, se um membro é marcado para expiração, o membro não expira a menos que o líder do grupo expire. No entanto, se um membro não estiver marcado para expiração e o líder do grupo expirar, então, todos os membros expiram.

Grupos de arquivos contêm apenas dados de backup e não podem conter dados do archive. Objetos de archive podem utilizar o campo **Descrição de Archive** para facilitar um tipo de agrupamento, se requerido por um aplicativo.

A chamada de **dsmGroupHandler** agrupa as operações. A função **dsmGroupHandler** deve ser chamada a partir de uma transação. A maioria das condições de erro do grupo é captada na chamada de **dsmEndTxn**l ou **dsmEndTxnEx**.

A estrutura `out` em **dsmEndTxnEx** inclui um novo campo, **groupLeaderObjId**. Esse campo contém o ID do objeto do líder do grupo se um grupo tiver sido aberto nessa transação. É possível criar um grupo em mais de uma transação. Um grupo não é confirmado nem salvo no servidor até um fechamento ser executado. **dsmGroupHandler** é uma interface que pode aceitar cinco operações diferentes. Elas incluem:

- DSM_GROUP_ACTION_OPEN
- DSM_GROUP_ACTION_CLOSE
- DSM_GROUP_ACTION_ADD
- DSM_GROUP_ACTION_ASSIGNTO
- DSM_GROUP_ACTION_REMOVE

Tabela 15 lista as ações da chamada de função **dsmGroupHandler**:

Tabela 15. funções dsmGroupHanlder

Ação	Descrição
OPEN	A ação OPEN cria um grupo. O próximo objeto enviado será o líder do grupo. O líder do grupo não pode ter conteúdo. Todos os objetos após o primeiro objeto tornam-se membros incluídos no grupo. Para criar um grupo, abra um grupo e transmita uma cadeia exclusiva para identificar o grupo. Esse identificador exclusivo permite que vários grupos com o mesmo nome sejam abertos. Após a abertura do grupo, o próximo objeto enviado é o líder do grupo. Todos os outros objetos enviados são membros do grupo.

Tabela 15. funções dsmGroupHanlder (continuação)

Ação	Descrição
CLOSE	A ação CLOSE confirma e salva um grupo aberto. Para fechar o grupo, transmita o nome do objeto e a cadeia exclusiva utilizada na operação de abertura. O aplicativo deve verificar por grupos abertos e, se necessário, fechar ou excluir os grupos. Um grupo não é confirmado ou salvo até que o grupo seja encerrado. Uma ação CLOSE falha nas condições a seguir: • O grupo que você está tentando fechar possui o mesmo nome que um grupo aberto existente. • Uma incompatibilidade de classe de gerenciamento existe entre o grupo encerrado atual e o novo grupo a ser fechado do mesmo nome. Neste caso, execute as etapas a seguir: 1. Consulte o grupo encerrado anterior. 2. Se a classe de gerenciamento do grupo encerrado existente for diferente da classe de gerenciamento associada ao grupo aberto atual, emita um **dsmUpdateObject** com o tipo DSM_BACKUPD_MC. Este comando atualiza o grupo existente para a nova classe de gerenciamento. 3. Emita a ação CLOSE.
ADD	A ação ADD anexa um objeto a um grupo. Todos os objetos que são enviados após a ação ADD são designados ao grupo.
ASSIGNTO	A ação ASSIGNTO permite que o cliente designe objetos existentes no servidor para o grupo de mesmo nível declarado. Essa transação configura o relacionamento do grupo PEER. A ação ASSIGNTO é semelhante à ação ADD, com as exceções a seguir: • A ação ADD se aplica a objetos em uma transação em andamento. • A ação ASSIGNTO se aplica a um objeto que está no servidor.
REMOVE	A ação REMOVE um membro ou uma lista de membros de um grupo. Um líder de grupo não pode ser removido de um grupo. Um membro do grupo deve ser removido antes de o membro poder ser excluído.

Use os tipos de consulta a seguir para suporte a grupo:
- **qtBackupGroups**
- **qtOpenGroups**

qtBackupGroups consultará grupos fechados, enquanto **qtOpenGroups** consultará grupos que estão abertos. O buffer da consulta para os novos tipos tem campos para **groupLeaderObjId** e **objType**. A consulta é executada de forma diferente, dependendo dos valores para esses dois campos. A tabela a seguir inclui algumas possibilidades de consulta:

Tabela 16. Exemplos de Consultas

groupLeaderObjId.hi	groupLeaderObjId.lo	objType	Resultado
0	0	NULL	Retorna uma lista de todos os líderes de grupos
grpLdrObjId.hi	grpLdrObjId.lo	0	Retorna uma lista de todos os membros do grupo designados ao líder do grupo especificado (**grpLdrObjId**).
grpLdrObjId.hi	grpLdrObjId.lo	objType	Retorna uma lista usando **BackQryRespEnhanced3**, para cada membro do grupo que é designado para o líder do grupo especificado (**grpLdrObjId**), e correspondendo o tipo de objeto (**objType**).

A estrutura da resposta (**qryRespBackupData**) de **dsmGetNextQObj** inclui dois campos para suporte ao grupo:

- **isGroupLeader**
- **isOpenGroup**

Estes campos são sinalizadores booleanos. O exemplo a seguir exibe a criação do grupo, incluindo membros para o grupo, e encerrando o grupo para confirmar o grupo no servidor IBM Spectrum Protect.

```
dsmBeginTxn
 dsmGroupHandler (PEER, OPEN, leader, uniqueId)
 dsmBeginSendObj
  dsmEndSendObj
dsmEndTxnEx (Com objId do líder)
Loop for multiple txns
{
 dsmBeginTxn
  dsmGroupHandler (PEER, ADD, member, groupLeaderObjID)
  Loop for multiple objects
  {
   dsmBeginSendObj
   Loop for data
   {
    dsmSendData
   }
   dsmEndSendObj
  }
 dsmEndTxn
}
dmBeginTxn
 dsmGroupHandler(CLOSE)
dsmEndTxn
```

Figura 15. Exemplo do Pseudocódigo que é Usado para Criar um Grupo

Para obter um exemplo de código, consulte o programa do grupo de amostra `dsmgrp.c` que é incluído no diretório `sampsrc` do API.

Recebendo Dados de um Servidor

Os aplicativos clientes podem receber dados ou objetos denominados e seus dados associados do armazenamento do IBM Spectrum Protect utilizando as funções de restauração e recuperação. A função de restauração acessa objetos dos quais foi feito backup anteriormente e a função de recuperação acessa objetos que foram anteriormente arquivados.

Restrição: A API pode restaurar ou recuperar somente objetos dos quais foi feito backup ou arquivados utilizando chamadas da API.

As funções de restauração e recuperação iniciam com uma operação de consulta. A consulta retorna diferentes informações, dependendo de se foi feito backup ou archive dos dados originalmente. Por exemplo, uma consulta sobre os objetos de backup retorna informações de se um objeto está ativo ou inativo, enquanto que uma consulta nos objetos de archive retorna informações como descrições de objetos. Ambas as consultas retornam IDs de objetos que são utilizados para identificar o objeto de forma exclusiva no servidor.

Restauração ou Recuperação Parcial de Objeto

O cliente aplicativo pode receber somente uma parte do objeto. Isso é chamado de uma restauração parcial do objeto ou uma recuperação parcial do objeto.

Atenção: A restauração ou recuperação parcial de objetos compactados ou criptografados produz resultados imprevisíveis.

Nota: Se você codifica seu aplicativo para utilizar uma restauração ou recuperação parcial de objeto, não pode compactar os dados ao enviá-lo. Para aplicar isso, configure *ObjAttr.objCompressed* como *bTrue*.

Para executar uma restauração ou recuperação parcial de objeto, associe os dois campos de dados a seguir a cada entrada **GetList** do objeto:

offset O deslocamento de bytes no objeto a partir do qual iniciar o retorno de dados.

length O número de bytes do objeto a retornar.

Utilize DSM_MAX_PARTIAL_GET_OBJ para determinar o número máximo de objetos que podem executar uma restauração ou recuperação parcial de objeto para uma lista **dsmBeginGetData** específica.

Os campos de dados a seguir, utilizados na chamada de **dsmBeginGetData**, determinam a parte do objeto que é restaurada ou recuperada:

- Se o deslocamento e o comprimento forem igual a zero, todo o objeto será restaurado ou recuperado do armazenamento do IBM Spectrum Protect.
- Se o deslocamento for maior do que zero, mas o comprimento for zero, o objeto será restaurado ou recuperado a partir do deslocamento até o final.
- Se o comprimento for maior do que zero, somente a parte do objeto a partir do deslocamento para o comprimento especificado será restaurada ou recuperada.

Restaurando ou Recuperando Dados

Após uma consulta ser feita e uma sessão ser estabelecida com o servidor IBM Spectrum Protect, você pode executar um procedimento para restaurar ou recuperar dados.

Para restaurar ou recuperar dados, conclua as seguintes etapas:

1. Consulte o servidor IBM Spectrum Protect para obter dados de backup ou archive.
2. Determinar os objetos a restaurar ou recuperar do servidor.
3. Classifique os objetos no campo **Ordem de Restauração**.
4. Enviar a chamada de **dsmBeginGetData** com a lista de objetos que você deseja acessar.
5. Envie a chamada **dsmGetObj** para obter cada objeto do sistema. Várias chamadas de **dsmGetData** podem ser necessárias para cada objeto obter todos os dados de objetos associados. Envie a chamada de **dsmEndGetObj** após todos os dados de um objeto serem obtidos.
6. Envie a chamada **dsmEndGetData** após todos os dados para todos os objetos serem recebidos ou para terminar a operação de recebimento.

Consultando o Servidor

Antes de iniciar qualquer operação de restauração ou recuperação, primeiro consulte o servidor IBM Spectrum Protect para determinar quais objetos podem ser recebidos do armazenamento.

Para enviar a consulta, o aplicativo deve inserir as listas e estruturas de parâmetros para a chamada **dsmBeginQuery**. A estrutura deve incluir o espaço no arquivo que a consulta examina e entradas de correspondência de padrões para os campos de nome de alto e baixo nível. Se a sessão tiver sido inicializada com um nome de proprietário NULL, não será necessário especificar o campo do proprietário. Entretanto, se uma sessão foi inicializada com um nome do proprietário explícito, apenas objetos que estão associados com esse nome do proprietário são retornados.

A consulta **BackupQuery** no momento fornece uma captura instantânea do sistema em um horário específico. Ao especificar uma data válida, é possível consultar todos os arquivos dos quais é feito backup naquele horário. Mesmo se um objeto tiver um backup ativo a partir de uma data posterior, o backup do ponto específico substitui o estado de um objeto de forma que a cópia inativa anterior seja retornada. Para obter mais informações, consulte o exemplo a seguir: pitDate.

Uma consulta retorna todas as informações que são armazenadas com o objeto, além das informações na tabela a seguir.

Tabela 17. Consultar as Informações de Retorno do Servidor

Campo	Descrição
`copyId`	Os valores `copyIdHi` e `copyIdLo` fornecem um número de 7 bytes que identifica exclusivamente este objeto para este nó no armazenamento do IBM Spectrum Protect. Utilize esse ID para solicitar um objeto específico a partir do armazenamento para o processo de restauração ou recuperação.
`restoreOrderExt`	O valor `restoreOrderExt` fornece um mecanismo para receber objetos do armazenamento do IBM Spectrum Protect na maneira mais eficiente possível. Classifique os objetos para restaurar esse valor para assegurar que as fitas sejam montadas somente uma vez e sejam lidas da frente para trás.

Você deve manter algumas ou todas as informações de consulta para processamento posterior. Mantenha os campos `copyId` e `restoreOrderExt`, pois são necessários para a operação de restauração real. Você também deve manter qualquer outra informação necessária para abrir um arquivo de dados ou identificar um destino.

Chame **dsmEndQuery** para concluir a operação de consulta.

Selecionando e Classificando Objetos pela Ordem de Restauração

Depois que a consulta de backup ou archive for executada, o cliente aplicativo deverá determinar quais objetos, se houver, devem ser restaurados ou recuperados.

Depois, classifique na ordem crescente (inferior para superior.) Essa classificação é muito importante para o desempenho da operação de restauração. A classificação dos objetos nos campos **restoreOrderExt** assegura que os dados sejam lidos a partir do servidor na ordem mais eficiente.

Todos os dados do disco são restaurados primeiro, seguidos por dados nas classes de mídia que requerem montagens de volumes (como fita). O campo **restoreOrderExt** também assegura que os dados na fita sejam lidos em ordem com o processamento começando no início de uma fita e progredindo em direção ao final.

A classificação apropriada no campo **restoreOrderExt** significa que montagens de fita duplicadas e retrocessos de fita desnecessários não ocorram.

Um valor diferente de zero no campo **restoreOrderExt**.top é correlacionado a um dispositivo de acesso serial exclusivo no servidor IBM Spectrum Protect. Como um dispositivo de acesso serial pode ser utilizado somente por uma sessão/ponto de montagem por vez, o aplicativo deve assegurar que se utilizar várias sessões, não ocorrerão restaurações simultâneas com o mesmo valor **restoreOrderExt**.top. Caso contrário, a primeira sessão poderá acessar os objetos, mas as outras sessões aguardarão até a primeira sessão encerrar e o dispositivo ser disponibilizado.

O exemplo a seguir mostra como classificar objetos usando os campos **Restaurar Ordem**.

Figura 16. Classificando Objetos com os Campos Restaurar Ordem

```
typedef struct {
dsStruct64_t      objId;
dsUint160_t      restoreOrderExt;

}  SortOrder;               /* estrutura utilizada para classificação */

=====================================================================
/* o código para classificação começa aqui */
dsmQueryType      queryType;
qryBackupData     queryBuffer;
DataBlk           qDataBlkArea;
qryRespBackupData qbDataArea;
dsInt16_t   rc;
dsBool_t  done = bFalse;
int i = 0;
int qry_item;
SortOrder  sortorder[100]; /* a classificação pode ser feita de até 100 itens
                                somente no momento.  Configure o
                                tamanho
                             da matriz apropriado para atender suas necessidades */
/*-------------------------------------------------------------------+
| NOTA: Certifique-se de que as inicializações apropriadas tenham
|  ocorrido para
queryType,
|       queryBuffer, qDataBlkAre e qbDataArea.
|
--------------------------------------------------------------------*/

   qDataBlkArea.bufferPtf = (char*) &qbDataArea;

   rc = dsmBeginQuery(dsmHandle, queryType, (void *) &queryBuffer);

   /*--------------------------------------+
   | Certifique-se de verificar rc de dsmBeginQuery
   +--------------------------------------*/
   while ( !done )
   {
      rc = dsmGetNextQObj(dsmHandle, &qDataBlkArea);
 if ((rc == DSM_RC_MORE_DATA) ||
         (rc == DSM_RC_FINISHED))
```

```
           &&( qDataBlkArea.numBytes))
      {
         /*****************************************/
         /* transferindo restoreOrderExt e objId */
         /*****************************************/
         sortorder[i].restoreOrderExt = qbDataArea.restoreOrderExt;
         sortorder[i].objId = qbDataArea.objId;

      } /*  if ((rc == DSM_RC_MORE_DATA) || (rc == DSM_RC_FINISHED)) */
      else
      {
          done = bTrue;
          /****************************/
          /* tome a ação apropriada. */
          /****************************/
      }

      i++;
      qry_item++;

   } /* while (!done) */
   rc = dsmEndQuery(dsmHandle);
  /*verifique rc  */
   /*****************************************************/
   /* classificando a matriz usando qsort.  Depois da chamada,   */
   /* sortorder será classificada pelo campo restoreOrderExt   */
   /*****************************************************/

   qsort(sortorder, qry_item, sizeof(SortOrder), SortRestoreOrder);

 /*-------------------------------------------------------------------+
| NOTA: Certifique-se de que os IDs de objetos classificados sejam
|       extraídos e armazenados
em qualquer estrutura de dados
|       desejada.
-------------------------------------------------------------------*/

/*-------------------------------------------------------------------+
| int SortRestoreOrder(SortOrder *a, SortOrder *b)
|
| Essa função compara os campos restoreOrder de  duas estruturas.
| if (a > b)
|   return(GREATERTHAN);
|| if (a < b)
|   return(LESSTHAN);
|| if (a == b)
|   return(EQUAL);
|+------------------------------------------------------------------*/
int SortRestoreOrder(SortOrder *a, SortOrder *b)
{
    if (a->restoreOrderExt.top > b->restoreOrderExt.top)
       return(GREATERTHAN);
    else if (a->restoreOrderExt.top < b->restoreOrderExt.top)
       return(LESSTHAN);
    else if (a->restoreOrderExt.hi_hi > b->restoreOrderExt.hi_hi)
       return(GREATERTHAN);
    else if (a->restoreOrderExt.hi_hi < b->restoreOrderExt.hi_hi)
       return(LESSTHAN);
    else if (a->restoreOrderExt.hi_lo > b->restoreOrderExt.hi_lo)
       return(GREATERTHAN);
    else if (a->restoreOrderExt.hi_lo < b->restoreOrderExt.hi_lo)
       return(LESSTHAN);
     else if (a->restoreOrderExt.lo_hi > b->restoreOrderExt.lo_hi)
       return(GREATERTHAN);
    else if (a->restoreOrderExt.lo_hi < b->restoreOrderExt.lo_hi)
       return(LESSTHAN);
    else if (a->restoreOrderExt.lo_lo > b->restoreOrderExt.lo_lo)
```

```
        return(GREATERTHAN);
    else if (a->restoreOrderExt.lo_lo < b->restoreOrderExt.lo_lo)
        return(LESSTHAN);
    else
        return(EQUAL);
}
```

Iniciando a Chamada de dsmBeginGetData

Depois de selecionar e classificar os objetos a serem recebidos, envie-os para o IBM Spectrum Protect para uma operação de restauração ou recuperação. A chamada de **dsmBeginGetData** inicia uma operação de restauração ou recuperação. Os objetos são retornados ao cliente aplicativo na ordem solicitada.

Complete as informações para esses dois parâmetros nessas chamadas:

mountWait
: Esse parâmetro informa ao servidor se o aplicativo cliente aguarda pela mídia off-line para montagem para obter dados de um objeto ou se esse objeto deve ser ignorado durante o processamento da operação de restauração ou recuperação.

dsmGetObjListP
: Esse parâmetro é uma estrutura de dados que contém o campo **objId** que é uma lista de todos os IDs de objetos que são restaurados ou recuperados. Cada **objId** é associado a uma estrutura **partialObjData** que descreve se todo o **objId** ou somente uma seção específica do objeto será recuperada.

  Cada **objId** tem oito bytes de comprimento, de forma que um único pedido de restauração ou recuperação possa conter milhares de objetos. O número de objetos a solicitar em uma única chamada está limitado a DSM_MAX_GET_OBJ ou DSM_MAX_PARTIAL_GET_OBJ.

Recebendo Cada Objeto para Restaurar ou Recuperar

Após o envio da chamada **dsmBeginGetData**, é possível executar um procedimento para receber cada objeto enviado a partir do servidor.

O código de retorno DSM_RC_MORE_DATA significa que um buffer foi retornado e que você deve chamar **dsmGetData** novamente. Verifique **DataBlk.numBytes** para obter o número real de bytes retornados.

Ao obter todos os dados para um objeto, você deve enviar uma chamada de **dsmEndGetObj**. Se mais objetos forem recebidos, envie **dsmGetObj** novamente.

Se desejar parar o processo, por exemplo, para descartar dados restantes no fluxo de restauração para todos os objetos que ainda não foram recebidos, envie a chamada **dsmEndGetData**. Esta chamada limpa os dados do servidor para o cliente. No entanto, a utilização desse método pode levar algum tempo para concluir. Se desejar terminar uma operação de restauração, use **dsmTerminate** para fechar a sessão.

1. Envie a chamada **dsmGetObj** para identificar o objeto solicitado do fluxo de dados e para obter o primeiro bloco de dados associado ao objeto.
2. Envie mais chamadas **dsmGetData**, conforme necessário, para obter os dados de objetos restantes.

Diagramas de Fluxo de Exemplo para Restauração e Recuperação

Um diagrama de estado e um fluxograma podem ser usados para ilustrar como executar operações de restauração ou recuperação.

A seta que aponta de "Em Objeto de Obtenção" para **dsmEndGetData** indica que é possível enviar uma chamada de **dsmEndGetData** após uma chamada a **dsmGetObj** ou **dsmGetData**. Pode ser necessário fazer isso caso ocorra uma condição de erro ao obter um objeto a partir do armazenamento de IBM Spectrum Protect e você queira parar a operação. Sob circunstâncias normais, chame **dsmEndGetObj** primeiro.

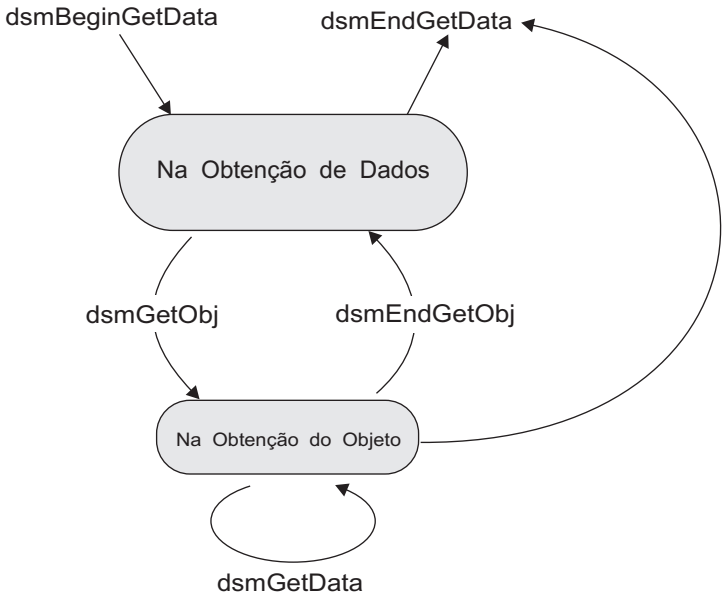


Figura 17. Diagrama de Estado para Operações de Restauração e Recuperação

A Figura 18 na página 74 exibe o fluxograma para executar operações de restauração ou recuperação.

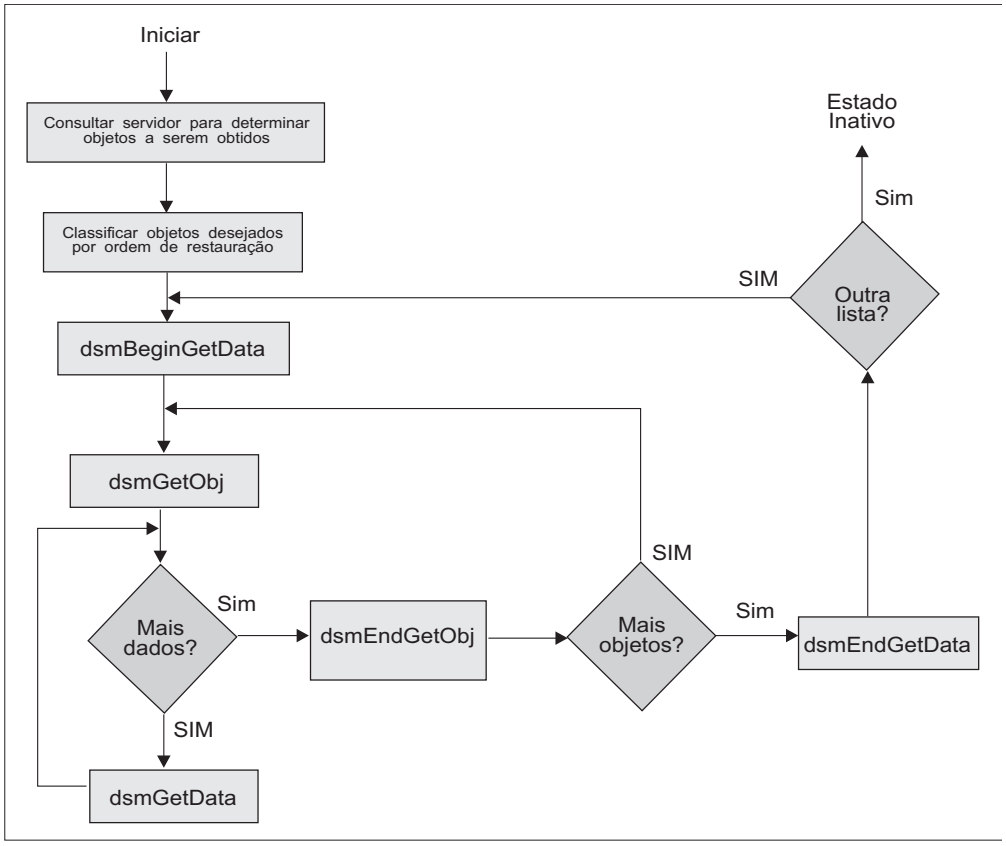


Figura 18. Fluxograma para Operações de Restauração e Recuperação

Exemplo de Código para Receber Dados de um Servidor

Esse exemplo demonstra a utilização das funções da API para recuperar dados do armazenamento do IBM Spectrum Protect.

A chamada de função **dsmBeginGetData** aparece dentro de uma instrução de comutação, de forma que diferentes parâmetros podem ser chamados, dependendo de se uma operação de restauração ou recuperação está sendo executada. A chamada de função **dsmGetData** é chamada a partir de um loop que obtém dados repetidamente do servidor até um sinalizador ser configurado, permitindo que a execução do programa saia do loop.

Figura 19. Um Exemplo de Recebimento de Dados de um Servidor

```
/* Chame dsmBeginQuery e crie uma lista vinculada de objetos para restaurar. */
/* Processe essa lista para criar a lista apropriada para as chamadas de GetData. */
/* Configure a estrutura getList para apontar para essa lista.                */
/* Esse exemplo é configurado para executar uma recuperação parcial do objeto.  Para   */
recuperar somente objetos completos, configure:                          */
/*       getList.stVersion = dsmGetListVersion;                             */
/*       getList.partialObjData = NULL;                                     */
dsmGetList getList;
getList.stVersion = dsmGetListPORVersion;   /* versão da estrutura   */
getList.numObjId  = items;               /* número de itens na lista */
getList.objId     = (ObjID *)rest_ibuff;
                                  /* lista de IDs de objetos a restaurar */
getList.partialObjData = (PartialObjData *) part_ibuff;
                                    /* lista de dados de objetos parciais */
```

```
switch(get_type)
{
   case (Restore_Get) :
     rc = dsmBeginGetData(dsmHandle,bFalse,gtBackup,&getList);
     break;
   case (Retrieve_Get) :
     rc = dsmBeginGetData(dsmHandle,bFalse,gtArchive,&getList);
     break;
   default : ;
}
if (rc)
{
   printf("*** dsmBeginGetData failed: ");
   rcApiOut(dsmHandle, rc);
   return rc;
}
/* Obtenha cada objeto da lista e verifique se está no
servidor /*.  Se estiver, inicialize as estruturas com os atributos do objeto para */
/* verificações de validação de dados.  Quando concluído, chame dsmGetObj.            */
 rc = dsmGetObj(dsmHandle,objId,&dataBlk);
done = bFalse;
while ( !done )
{
   if (   (rc == DSM_RC_MORE_DATA)
       || (rc == DSM_RC_FINISHED))
   {
      if (rc == DSM_RC_MORE_DATA)
      {
         dataBlk.numBytes = 0;
         rc = dsmGetData(dsmHandle,&dataBlk);
      }
      else
         done = bTrue;
   }
   else
   {
      printf("*** dsmGetObj or dsmGetData failed: ");
      rcApiOut(dsmHandle, rc);
      done = bTrue;
   }
} /* while */
rc = dsmEndGetObj(dsmHandle);
/* verifique rc de dsmEndGetObj  */
/* verifique rc de dsmEndGetData */
rc = dsmEndGetData(dsmHandle);
return 0;
```

Atualizando e Excluindo Objetos no Servidor

Os aplicativos da API podem utilizar a chamada de função **dsmUpdateObj** ou **dsmUpdateObjEx** para atualizar objetos que foram arquivados ou dos quais foi feito backup. Utilize qualquer uma das chamadas apenas no estado de sessão, atualizando um objeto por vez. Utilize **dsmUpdateObjEx** para atualizar qualquer um dos vários objetos de archive que contêm o mesmo nome.

Para selecionar um objeto de archive, configure a chamada de função **dsmSendType** para **stArchive**.

- Com **dsmUpdateObj**, apenas o objeto de archive mais recente com o nome designado é atualizado.
- Com **dsmUpdateObjEx**, qualquer objeto arquivado pode ser atualizado especificando o ID de objeto adequado.

Para um objeto arquivado, o aplicativo pode atualizar os seguintes campos:

- Descrição
- Informações do objeto
- Owner

Para selecionar um objeto de backup, configure **dsmSendType** para **stBackup**. Para objetos com backup, somente a cópia ativa é atualizada.

Para um objeto com backup, o aplicativo pode atualizar os seguintes campos:

- Classe de Gerenciamento
- Informações do objeto
- Owner

Excluindo Objetos do Servidor

Os aplicativos da API podem fazer chamadas para excluir objetos que foram arquivados ou desativar objetos dos quais foi feito backup. A exclusão de objetos arquivados depende da autorização do nó que foi fornecida quando o administrador registrou o nó. Os administradores podem especificar que os nós podem excluir objetos arquivados.

Utilize a chamada de função **dsmDeleteObj** para excluir objetos arquivados e desativar os objetos de backup. A utilização de **delType** remove o objeto de backup do servidor. Isso é baseado em **objID**, exclui um objeto do banco de dados do servidor. Somente o proprietário de um objeto pode excluí-lo. É possível excluir qualquer versão (ativa ou inativa) de um objeto. O servidor reconcilia as versões. Se você excluir uma versão ativa de um objeto, a primeira versão inativa torna-se ativa. Se você excluir uma versão inativa de um objeto, todas as versões mais antigas serão avançadas. O nó deve ser registrado com a permissão **backDel**.

Um objeto arquivado é marcado para exclusão no armazenamento quando o sistema executa seu próximo ciclo de expiração do objeto. Após a exclusão de um objeto arquivado a partir do servidor, não será possível recuperá-lo.

Ao deixar um objeto de backup inativo no servidor, o objeto vai de um estado ativo para um estado inativo. Esses estados têm políticas de retenção diferentes associadas a eles que são baseadas na classe de gerenciamento designada.

Semelhante à chamada de **dsmSendObj**, uma chamada a **dsmDeleteObj** é enviada dentro dos limites de uma transação. O diagrama de estado na Figura 12 na página 61 exibe como uma chamada a **dsmDeleteObj** é precedida por uma chamada a **dsmBeginTxn** e seguida por uma chamada a **dsmEndTxn**.

Criando Log de Eventos

Um aplicativo da API pode registrar mensagens de eventos em locais centrais. O aplicativo pode direcionar a criação de log para o servidor IBM Spectrum Protect, para a máquina local, ou para ambos. A chamada de função **dsmLogEventEx** é executada em uma sessão. Para visualizar mensagens registradas no servidor, use o comando **actlog** da consulta através do cliente administrativo.

Use a opção do cliente do IBM Spectrum Protect, `errorlogretention`, para remover o arquivo do log de erros do cliente se o aplicativo gravar várias mensagens do cliente no log do cliente `dsmLogType`, `logLocal` ou `logBoth`.

Para obter mais informações sobre logs do IBM Spectrum Protect, consulte a documentação do servidor do IBM Spectrum Protect.

Resumo do diagrama de estado para a API do IBM Spectrum Protect

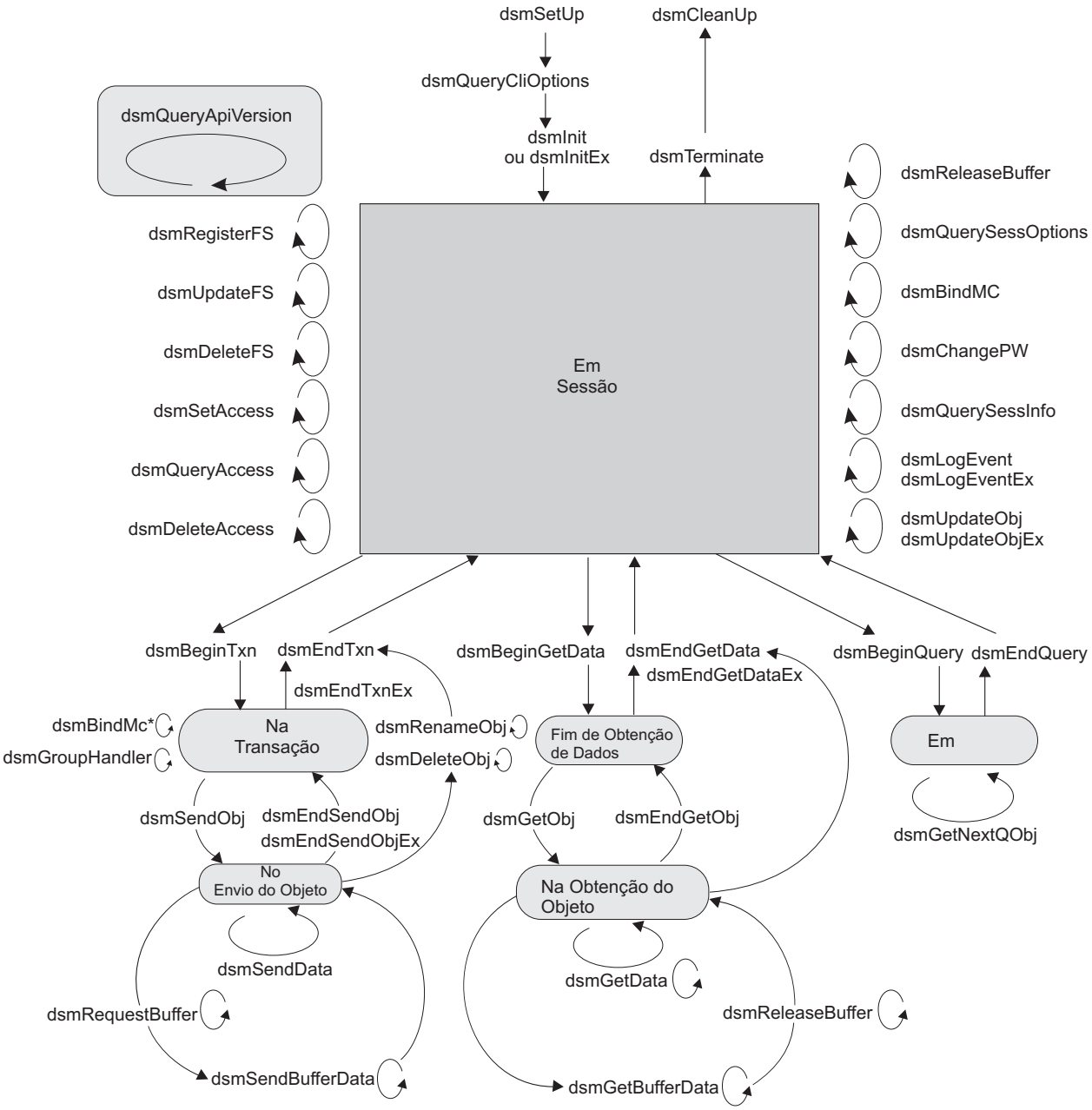
Após revisar todas as considerações para criar seu próprio aplicativo com a API do IBM Spectrum Protect, revise este resumo do diagrama de estado de um aplicativo inteiro.

A Figura 20 na página 78 contém o diagrama de estado para a API. Contém todos os diagramas de estado anteriormente exibidos, além de várias outras chamadas não exibidas anteriormente.

Os pontos deste diagrama incluem:

- Chame **dsmQueryApiVersionEx** a qualquer momento. Não tem nenhum estado associado a ela. Consulte a Figura 1 na página 15 para obter um exemplo.
- Chame **dsmQueryCliOptions** somente antes de uma chamada **dsmInitEx**.
- Utilize **dsmRegisterFS**, **dsmUpdateFS** e **dsmDeleteFS** para gerenciar espaços no arquivo. Essas chamadas são feitas a partir de um estado de sessão inativo. Utilize a chamada de **dsmBeginQuery** para consultar espaços no arquivo. Para obter informações adicionais sobre chamadas de espaço no arquivo, consulte "Gerenciando Espaços no Arquivo" na página 26.
- Envie a chamada de **dsmBindMC** a partir de um estado de sessão inativo ou de um estado de transação de objeto de envio. Consulte o exemplo na Figura 8 na página 31.
- Envie a chamada de **dsmChangePW** a partir de um estado de sessão inativo.

  Nota: Se a chamada de **dsmInitEx** retornar com um código de retorno de senha expirada, a chamada de **dsmChangePW** deve ser realizada antes do início de uma sessão válida. Consulte a Figura 4 na página 21 para obter um exemplo que utiliza **dsmChangePW**.
- Se uma chamada retornar com um erro, o estado permanece como estava. Por exemplo, se **dsmGetObj** retornar com um erro, o estado permanecerá Em Obtenção de Dados e uma chamada para **dsmEndGetObj** será um erro de seqüência de chamada.



* Pode ser dentro ou fora de uma

Figura 20. Diagrama de Estado de Resumo para a API

Capítulo 4. Entendendo a Interoperabilidade

A API tem dois tipos de interoperabilidade: entre o cliente de backup-archive e os aplicativos da API e entre diferentes sistemas operacionais.

Interoperabilidade do Cliente de Backup/Archive

A linha de comandos de backup/archive pode acessar objetos de API para fornecer interoperabilidade limitada. Objetos da API só podem ser visualizados e acessados a partir do cliente da linha de comandos de backup-archive e não podem ser visualizados ou acessados de qualquer uma das interfaces gráficas. O cliente da linha de comandos de backup-archive só pode restaurar o conteúdo do arquivo e nada mais, para que você só use-o para um tipo de operação de salvamento.

As ações a seguir da linha de comandos são fornecidas:
- Excluir archive
- Excluir espaço no arquivo
- Consultar
- Restaurar
- Recuperar
- Configurar acesso

As informações de caminho são os diretórios em si para os objetos de cliente de backup/archive. Em contraste, as informações do caminho do objeto da API podem não ter nenhum relacionamento com diretórios existentes: o caminho pode ser totalmente inventado. A interoperabilidade não altera esse aspecto desses tipos de objetos. Para utilizar esse recurso com êxito, siga as restrições e convenções.

Notas:

1. Não há nenhuma interoperabilidade entre o cliente de backup/archive e os objetos da API armazenados em um servidor de proteção de retenção.
2. Você não pode usar as GUIs do cliente de backup-archive para acessar arquivos armazenados usando o cliente de API. Você só pode usar a linha de comandos para acessar esses arquivos.

Nomeando os Objetos da API

Estabeleça uma convenção de nomenclatura consistente para os nomes do objeto da API. A convenção de nomenclatura deve fornecer o nome do espaço no arquivo, o qualificador de alto nível e o qualificador de baixo nível. O nome do espaço no arquivo e qualificadores de alto nível podem se referir a nomes reais do diretório. Cada nome do objeto pode consistir de mais de um nome do diretório que se aplique ao qualificador de baixo nível.

Por conveniência, use o nome do objeto que não está prefixado com informações do diretório como o qualificador de baixo nível. Para obter informações adicionais, consulte “IDs e Nomes de Objetos” na página 23.

Os nomes do espaço no arquivo devem ser totalmente qualificados quando eles forem referidos a partir da APLI ou da linha de comandos do backup-archive. Por exemplo, em um sistema operacional UNIX ou Linux, você registra os espaços no arquivo a seguir:

- /a
- /a/b

Quando se referir a /a, os objetos que estiverem relacionados apenas ao espaço no arquivo /a são exibidos. Para visualizar objetos que estão relacionados a /a/b, você deve especificar /a/b como o nome do espaço no arquivo.

Depois de registrar ambos os espaços no arquivo, se fizer backup do objeto b no espaço no arquivo/a, então uma consulta para /a/b continua a exibir objetos que são relacionados apenas ao espaço no arquivo /a/b.

A exceção a essa restrição ocorre nas referências do espaço no arquivo quando você tenta consultar ou excluir espaços no arquivo com a API. Em ambos os casos, os nomes do espaço no arquivo não têm que ser totalmente qualificados se você usar um caractere curinga. Por exemplo, /a* se refere a ambos /a e /a/b.

Dica: Se uma interoperabilidade for importante para você, então evite nomes de espaço no arquivo que se sobreponham.

Em sistemas Windows, coloque os nomes de espaços no arquivo entre chaves { } para objetos da API quando acessar os objetos a partir da interface da linha de comandos de backup-archive. Sistemas operacionais Windows automaticamente colocam nomes de espaço no arquivo em letras maiúsculas quando você registra ou se refere aos nomes. No entanto, essa função automática não ocorre para o restante da especificação do nome do objeto. Se quiser interoperabilidade completa, coloque o qualificador de nível superior e o qualificador de nível inferior em letras maiúsculas no aplicativo ao fazer backup dos objetos da API. Se seu aplicativo não converter em maiúsculas os qualificadores de alto nível (nomes de diretórios) e os qualificadores de baixo nível (nomes de arquivos) antes de enviar objetos para o servidor, não será possível acessar os objetos diretamente por nome, por meio do cliente de backup-archive.

Por exemplo, se um objeto estiver armazenado no servidor como {"FileSpacename"}\TEST\MYDIRNAME\file.txt, não será possível restaurar ou consultar o objeto file.txt diretamente, porque seu aplicativo não converteu em maiúsculas o nome do arquivo antes de o arquivo ser copiado para o servidor. A única maneira de manipular esses objetos é usar caracteres curinga. Por exemplo, para consultar \TEST\MYDIRNAME\file.txt, um usuário do cliente de backup-archive deve usar caracteres curinga para todas as partes do nome do objeto que não foram convertidas em maiúsculas antes de serem enviadas para o servidor. O comando a seguir deve ser usado para consultar este arquivo file.txt:

```
dsmc query backup {"FileSpaceName"}\TEST\MYDIRNAME\*
```

Se alguns outros dos outros qualificadores também forem salvos com texto em minúsculas, esses qualificadores também deverão ser consultados usando curingas. Por exemplo, para consultar um objeto que está armazenado como {"FileSpaceName"}\TEST\mydirname\file.txt, use o seguinte comando:

```
dsmc query backup {"FileSpaceName"}\TEST\*\*
```

Os exemplos a seguir demonstram esses conceitos. Em ambos os ambientes Windows e UNIX ou Linux, você não tem que especificar o qualificador de alto ou baixo nível. Entretanto, se não especificar p qualificador completo, então você deve usar o caractere curinga.

Plataforma	Exemplo
Windows	Para consultar todos os arquivos de backup no espaço no arquivo MYFS, insira a sequência a seguir: `dsmc q ba "{MYFS}\*\*"` Você deve usar pelo menos um asterisco (*) para cada um dos qualificadores de alto e de baixo nível.
UNIX ou Linux	Para consultar todos os arquivos de backup no espaço no arquivo /A, insira a sequência a seguir: `dsmc q ba "/A/*/*"` Você deve usar pelo menos um asterisco (*) para cada um dos qualificadores de alto e de baixo nível.

Comandos do Cliente de Backup-Archive que Você Pode Usar com a API

Você pode usar um subconjunto de comandos do cliente de backup-archive em um aplicativo. Por exemplo, você pode visualizar e gerenciar objetos que outros usuários possuem no mesmo nó ou em um nó diferente.

Para visualizar e gerenciar objetos que outros usuários possuem no mesmo nó ou em um nó diferente, execute estas etapas:

1. Forneça acesso com o comando **set access**.
2. Especifique o proprietário e o nó. Utilize as opções *fromowner* e *fromnode* a partir da linha de comandos de backup/archive para especificar o proprietário e o nó. Por exemplo:

```
dsmc q ba "/A/*/*"  -fromowner=other_owner  -fromnode=other_node
```

A Tabela 18 descreve os comandos que podem ser utilizados com os objetos da API.

Tabela 18. Comandos do Cliente de Backup-Archive que Você Pode Usar com Objetos da API

Comando	Descrição
Delete Archive	Os arquivos arquivados possuídos pelo usuário atual podem ser excluídos. As configurações do comando set access não têm efeito neste comando.
Delete Filespace	O comando **delete filespace** afeta os objetos da API.
Query	Na linha de comandos de backup-archive, é possível consultar objetos da API de backup e arquivados e objetos que outros usuários possuem, ou que existem em outros nós. Consulte "Nomeando os Objetos da API" na página 79 para obter informações sobre como consultar objetos da API. Use a opção *–fromowner* existente para consultar objetos que um usuário diferente possui para o qual a permissão set access foi fornecida. Use a opção existente *-fromnode* para consultar objetos que existem em outro nó para o qual a permissão set access foi fornecida. Para obter informações adicionais, consulte "**dsmInitEx**" na página 121.

Tabela 18. Comandos do Cliente de Backup-Archive que Você Pode Usar com Objetos da API (continuação)

Comando	Descrição
Restauração **Recuperar**	**Nota:** Use esses comandos apenas para situações de exceção. Os objetos da API que forem criptografados usando a chave gerenciada por aplicativo podem ser restaurados ou recuperados, se a chave de criptografia for conhecida ou salva no arquivo de senhas ou no registro. Os objetos da API criptografados usando a criptografia transparente não podem ser restaurados ou recuperados usando o cliente de backup-archive. Esses comandos retornam dados como arquivos de bits que são criados usando atributos de arquivo padrão. É possível restaurar ou recuperar objetos da API possuídos por outros usuários ou que são de um nó diferente. O comando set access determina quais objetos qualificar.
Set Access	O comando **set access** permite que os usuários gerenciem objetos da API possuídos por outros usuários ou que são de outro nó.

Interoperabilidade do Sistema Operacional

A API do IBM Spectrum Protect suporta interoperabilidade multiplataforma. Os aplicativos em um sistema UNIX ou Linux podem operar em espaços no arquivo e em objetos dos quais é feito backup a partir de um sistema Windows. Similarmente, um sistema Windows pode operar em espaços no arquivo e em objetos nos quais é feito backup a partir de um sistema UNIX ou Linux.

Por padrão, os nomes dos objetos de um sistema UNIX são compatíveis com os nomes de objetos de outros sistemas UNIX. Por padrão, nomes dos objetos de sistemas Windows não são compatíveis com nomes de objetos de sistemas UNIX. Vários parâmetros controlam a nomenclatura de objetos nos espaços no arquivo IBM Spectrum Protect. Se você configurar um aplicativo de forma apropriada, os nomes de objetos poderão ser usados por aplicativos que são executados em sistemas Windows e UNIX. Use os mesmos parâmetros para fazer backup e restaurar objetos.

Restrição: Um aplicativo Windows que usa Unicode cria um espaço no arquivo que não é compatível com aplicativos executados em sistemas UNIX.

Para obter interoperabilidade, conclua as seguintes tarefas de configuração:

1. Estabeleça uma convenção de nomenclatura consistente. Selecione um caractere para o delimitador `dir`, como barra (`/`) ou barra invertida (`\`). Coloque o caractere de delimitador de diretório em frente ao nome do espaço no arquivo, o qualificador de alto nível e o qualificador de baixo nível.
2. Quando chamar **dsmInitEx**, configure o valor do campo **dirDelimiter** para o caractere do delimitador de diretório que você selecionou e configure **bCrossPlatform** para **bTrue**.
3. Configure o sinalizador **useUnicode** para **bFalse** quando você usar a interface IBM Spectrum Protect. Nomes de arquivo Unicode não são compatíveis com nomes de arquivo não Unicode.

Fazendo Backup de Vários Nós com Suporte ao Proxy de Nó Cliente

Backups de diversos nós que compartilham armazenamento podem ser consolidados em um nome do nó de destino comum no servidor IBM Spectrum Protect. Este método é útil quando o sistema que executa o backup puder alterar ao longo do tempo, como com um cluster. Também é possível usar a opção `asnodename` para restaurar dados de um sistema diferente daquele que executou o backup.

Use a opção `asnodename` na sequência de opções **`dsmInitEx`** para fazer backup, arquivar, restaurar, recuperar, consultar ou excluir dados no nome do nó de destino no servidor IBM Spectrum Protect. Também é possível especificar a opção `asnodename` no arquivo `dsm.opt` ou `dsm.sys`.

Restrição: Não use nós de destino como nós tradicionais, especialmente se você criptografar seus arquivos antes de fazer backup para o servidor.

Para ativar esta opção, conclua as etapas a seguir:

1. Instale o cliente da API em todos os nós em um ambiente de dados compartilhado.
2. Se ainda não estiver registrado, registre cada nó com o servidor IBM Spectrum Protect. Registre o nome do nó de "destino" comum a ser compartilhado por cada um dos nós do agente que são usados em seu ambiente de dados compartilhado.
3. Registre cada um dos nós do agente no ambiente de dados compartilhados com o servidor. O nome do nó do agente é usado para autenticação. Os dados não são armazenados usando o nome do nó do agente quando a opção `asnodename` é usada.
4. Peça ao administrador que conceda autoridade de proxy para todos os nós no ambiente compartilhado para acessar o nome do nó de destino no servidor IBM Spectrum Protect, utilizando **`grant proxynode`** .
5. Use o comando do cliente administrativo **`query proxynode`** para exibir os nós clientes que têm a autoridade para executar operações de cliente em nome de outro nó. Esta autoridade é concedida pelo comando **`grant proxynode`**. Ou use o comando **`dsmQuery`** com o tipo de consulta **`qtProxyNodeAuth`** para ver os nós para os quais o nó pode realizar proxy.
6. Se o aplicativo estiver usando criptografia de dados, não `TSMENCRKEY`, assegure-se de que todos os nós usam a mesma chave de criptografia. Você deve usar a mesma chave de criptografia para todos os arquivos para os quais é feito backup no ambiente do nó compartilhado.

Tarefas relacionadas:

Fazendo backup de dados com suporte ao proxy do nó cliente (sistemas UNIX e Linux)

Fazendo backup de dados com suporte ao proxy do nó cliente (sistemas Windows)

Capítulo 5. Utilizando a API com Unicode

A API do IBM Spectrum Protect API suporta Unicode UCS2, uma página de códigos de byte duplo e comprimento fixo que possui pontos de código para todas as páginas de códigos conhecidas, como japonês, chinês ou alemão. Suporta até 65.535 pontos de código exclusivos.

Restrição: Este recurso está disponível somente no Windows.

Com Unicode, seu aplicativo pode fazer backup e restaurar nomes de arquivos em qualquer conjunto de caracteres a partir da mesma máquina. Por exemplo, em uma máquina em inglês, é possível fazer backup e restaurar nomes de arquivos na página de código de qualquer outro idioma.

Quando Usar Unicode

É possível simplificar o aplicativo que suporta vários idiomas compondo um aplicativo Unicode e obtendo vantagem da interface Unicode do IBM Spectrum Protect.

Use a interface Unicode do IBM Spectrum Protect se qualquer uma das condições a seguir for verdadeira:

- Se seu aplicativo já estiver compilado para Unicode e estava convertendo para um conjunto de caracteres multibyte (mbcs) antes de chamar a API do IBM Spectrum Protect.
- Se estiver gravando um novo aplicativo e desejar ativar seu aplicativo para suportar Unicode.
- Se seu aplicativo utilizar uma cadeia transmitida a ele do sistema operacional ou de outro aplicativo que utiliza Unicode.

Se não precisar de Unicode, não é necessário compilar seu aplicativo novamente.

A API continua a suportar a interface dsm. O SDK da API contém os programas de amostra callmtu1.c e callmtu2.c que demonstram como utilizar a API Unicode. Use **`makemtu`** para compilar esses programas.

Configurando Unicode

Para configurar e usar Unicode, você deve desempenhar um procedimento específico para que a API registre um espaço no arquivo Unicode no servidor e todos os nomes de arquivos do espaço no arquivo tornem-se cadeias Unicode.

Restrição: Não é possível armazenar nomes de arquivo Unicode e não-Unicode no mesmo espaço no arquivo.

1. Compile o código com o sinalizador `-DUNICODE`.
2. Todas as cadeias de seu aplicativo devem ser cadeias wchar.
3. Siga as estruturas do arquivo tsmapitd.h e as definições de função no arquivo tsmapifp.h para chamadas à API.
4. Configure o sinalizador *`useUnicode`* para *`bTrue`* na chamada de função **`tsmInitEx`**. Qualquer novo espaço no arquivo será registrado como um espaço no arquivo Unicode.

Ao enviar dados para espaços no arquivo não-Unicode registrados, a API continua a enviar nomes de arquivos como não-Unicode. Renomeie os espaços no arquivo antigos no servidor para fsname_old e inicie um novo espaço no arquivo Unicode para novos dados. A API restaura os dados não-Unicode a partir antigos espaços no arquivo. utilize o campo **bIsUnicode** na estrutura **tsmQryRespFSData** retornada em um espaço no arquivo de consulta para determinar se um espaço no arquivo é ou não Unicode.

Cada chamada de função **dsmXXX** tem uma chamada de função **tsmXXX** correspondente. A diferença entre as duas são as estruturas utilizadas. Todas as estruturas da chamada de função **tsmXXX** têm tipos dsChar_t para valores de sequência quando compiladas com a sinalização UNICODE. dsChar_r é mapeado para wchar. Não há nenhuma outra diferença entre essas interfaces.

Restrição: Utilize uma das duas interfaces. Não combine as interfaces da chamada de função **dsmXXX** e da chamada de função **tsmXXX**. Assegure-se de usar as estruturas do IBM Spectrum Protect e as definições de versão do IBM Spectrum Protect.

Algumas constantes continuam a ser definidas no arquivo dsmapitd.h, portanto, você precisará dos arquivos dsmapitd.h e tsmapitd.h ao compilar.

É possível usar a interface do IBM Spectrum Protect em outros sistemas operacionais, como UNIX ou Linux, mas nesses sistemas operacionais, o tipo dsChar_t é mapeado para char, porque o Unicode é suportado somente em sistemas operacionais Windows. Você pode gravar somente uma variação do aplicativo e compilar mais de um sistema operacional usando a interface IBM Spectrum Protect. Se estiver gravando um novo aplicativo, use a interface IBM Spectrum Protect.

Se estiver fazendo upgrade de um aplicativo existente:
1. Converta as estruturas e chamadas da chamada de função **dsmXXX** para a interface IBM Spectrum Protect.
2. Migre espaços existentes no arquivo.
3. Faça backup de novos espaços no arquivo com o sinalizador *useUnicode* configurado para *true*.

Nota: Após utilizar um cliente ativado para Unicode para acessar um nó, não é possível conectar ao mesmo nome de nó com uma versão mais antiga da API ou com uma API de outro sistema operacional. Se seu aplicativo utiliza capacidade entre plataformas, não utilize o sinalizador `Unicode`. Não há nenhum suporte entre plataformas entre os sistemas operacionais Unicode e não-Unicode.

Ao ativar o sinalizador *useUnicode*, todas as estruturas de cadeia são tratadas como cadeias Unicode. No servidor, somente os seguintes campos são verdadeiramente Unicode:
- Nome do Espaço no Arquivo
- Nível superior
- Nível inferior
- Descrição de archive

Todos os demais campos são convertidos para mbcs na página de códigos local antes de serem enviados ao servidor. Os campos, como nome do nó, são cadeias wchar. Devem ser válidos no código do idioma atual. Por exemplo, em uma máquina em japonês, é possível fazer backup de arquivos com nomes em chinês, mas o nome do nó deve ser uma cadeia válida em japonês. O arquivo de opções

permanece na página de códigos atual. Se você precisar criar uma lista de inclusão-exclusão Unicode, use a opção *inclexcl* com um nome de arquivo e crie um arquivo Unicode com padrões Unicode.

Referências relacionadas:

Opção inclexcl

Capítulo 6. Chamadas de Função da API

O Tabela 19 fornece uma lista alfabética das chamadas de funções de API, uma breve descrição e o local de informações mais detalhadas sobre a chamada de função, que inclui:

Elemento	Descrição
Finalidade	Descreve a chamada da função.
Sintaxe	Contém o código C em si para a chamada de função. Este código é copiado da versão do UNIX ou Linux do arquivo de cabeçalho dsmapifp.h. Consulte a seção Apêndice C, "Arquivo de Origem de Definições de Função da API", na página 207. Este arquivo é ligeiramente diferente em outros sistemas operacionais. Os programadores de aplicativos para outros sistemas operacionais devem verificar sua versão do arquivo de cabeçalho dsmapifp.h para obter a sintaxe exata das definições de API.
Parâmetros	Descreve cada parâmetro da chamada de função, identificando-o como entrada (E) ou saída (S), dependendo de como é utilizado. Alguns parâmetros são designados como ambos, entrada e saída (E/S). Os tipos de dados que são referidos nesta seção são definidos no arquivo de cabeçalho dsmapitd.h. Consulte a seção Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163.
Códigos de Retorno	Contém uma lista dos códigos de retorno que são específicos para a chamada de função. Erros gerais do sistema, como erros de comunicação, problemas de servidor ou erros do usuário, que podem aparecer em qualquer chamada não são listados. Os códigos de retorno são definidos no arquivo de cabeçalho dsmrc.h. Consulte o Apêndice A, "Arquivo de Origem dos Códigos de Retorno do API : `dsmrc.h`", na página 153.

Tabela 19. Chamadas de Função da API

Chamada de função e local	Descrição
"dsmBeginGetData" na página 91	Inicia uma operação de restauração ou recuperação em uma lista de objetos em armazenamento.
"dsmBeginQuery" na página 93	Inicia uma solicitação de consulta ao IBM Spectrum Protect para obter informações.
"dsmBeginTxn" na página 98	Inicia uma ou mais transações que inicial uma ação completa. Todas as ações foram bem-sucedidas ou nenhuma foi bem-sucedida.
"dsmBindMC" na página 99	Associa ou liga uma classe de gerenciamento ao objeto que é transmitido.
"dsmChangePW" na página 100	Muda uma senha do IBM Spectrum Protect.
"dsmCleanUp" na página 101	Essa chamada é utilizada se **`dsmSetUp`** tiver sido chamado.
"dsmDeleteAccess" na página 101	Exclui as regras de autorização atuais para versões de backup ou cópias arquivadas de seus objetos.
"dsmDeleteFS" na página 102	Exclui um espaço no arquivo do armazenamento.
"dsmDeleteObj" na página 103	Desativa objetos de backup ou exclui objetos de archive do armazenamento.

Tabela 19. Chamadas de Função da API (continuação)

Chamada de função e local	Descrição
"dsmEndGetData" na página 104	Encerra uma sessão de **dsmBeginGetData** que obtém objetos do armazenamento.
"dsmEndGetDataEx" na página 105	Fornece o total de bytes sem a LAN que foram enviados.
"dsmEndGetObj" na página 105	Encerra uma sessão de **dsmGetObj** que obtém dados para um objeto especificado.
"dsmEndQuery" na página 106	Significa o final de uma ação de **dsmBeginQuery**.
"dsmEndSendObj" na página 106	Indica o final dos dados que são enviados ao armazenamento.
"dsmEndSendObjEx" na página 107	Fornece informações de compactação e o número de bytes que foram enviados.
"dsmEndTxn" na página 108	Termina uma transação do IBM Spectrum Protect.
"dsmEndTxnEx" na página 109	Fornece informações do ID do objeto líder do grupo que serão utilizadas com a chamada de **dsmGroupHandlerfunction**.
"dsmGetData" na página 110	Obtém um fluxo de bytes de dados do IBM Spectrum Protect e os coloca no buffer do responsável pela chamada.
"dsmGetBufferData" na página 111	Obtém um buffer de dados alocado pelo IBM Spectrum Protect a partir do servidor IBM Spectrum Protect.
"dsmGetNextQObj" na página 112	Obtém a resposta da próxima consulta de uma chamada anterior de **dsmBeginQuery** e coloca a mesma no buffer do responsável pela chamada.
"dsmGetObj" na página 115	Obtém os dados do objeto solicitado do fluxo de dados e coloca-os no buffer do responsável pela chamada.
"dsmGroupHandler" na página 116	Executa uma ação em um grupo de arquivos lógico, dependendo da entrada fornecida.
"dsmInit" na página 117	Inicia uma sessão de API e conecta o cliente ao armazenamento.
"**dsmInitEx**" na página 121	Inicia uma sessão de API utilizando os parâmetros adicionais que permitem verificação estendida.
"dsmLogEvent" na página 125	Registra uma mensagem de usuário no arquivo de log do servidor, no log de erros local ou em ambos.
"dsmLogEventEx" na página 126	Registra uma mensagem de usuário no arquivo de log do servidor, no log de erros local ou em ambos.
"dsmQueryAccess" na página 127	Consulta no servidor todas as regras de autorização de acesso para versões de backup ou cópias arquivadas de seus objetos.
"dsmQueryApiVersion" na página 128	Executa um pedido de consulta à versão da biblioteca de API que o cliente aplicativo acessa.
"dsmQueryApiVersionEx" na página 128	Executa um pedido de consulta à versão da biblioteca de API que o cliente aplicativo acessa.
"dsmQueryCliOptions" na página 129	Consulta valores de opções importantes nos arquivos de opções do usuário.
"dsmQuerySessInfo" na página 130	Inicia um pedido de consulta no IBM Spectrum Protect para obter informações que são relacionadas à operação da sessão especificada em **dsmHandle**.

Tabela 19. Chamadas de Função da API (continuação)

Chamada de função e local	Descrição
"dsmQuerySessOptions" na página 131	Consulta valores de opções importantes que são válidos na sessão especificada em **dsmHandle**.
"dsmRCMsg" na página 132	Obtém o texto de mensagem associado a um código de retorno de API.
"dsmRegisterFS" na página 133	Registra um novo espaço no arquivo no servidor.
"dsmReleaseBuffer" na página 134	Retorna um buffer alocado pelo IBM Spectrum Protect.
"dsmRenameObj" na página 134	Renomeia o nome do objeto de nível superior ou de nível inferior.
"dsmRequestBuffer" na página 136	Obtém um buffer alocado pelo IBM Spectrum Protect para eliminação da cópia de buffer.
"dsmRetentionEvent" na página 137	Envia uma lista de IDs de objetos ao servidor com uma operação de evento de retenção a ser realizada nesses objetos.
"dsmSendBufferData" na página 138	Envia dados de um buffer alocado pelo IBM Spectrum Protect.
"dsmSendData" na página 139	Envia um fluxo de byte de dados ao IBM Spectrum Protect por meio de um buffer.
"dsmSendObj" na página 140	Inicia um pedido para enviar um único objeto ao armazenamento.
"dsmSetAccess" na página 144	Fornece a outros usuários ou nós acesso a versões de backup ou cópias arquivadas de seus objetos, acesso a todos os seus objetos ou acesso a um conjunto seletivo.
"dsmSetUp" na página 145	Sobrescreve os valores de variáveis de ambiente.
"dsmTerminate" na página 147	Encerra uma sessão com o servidor e limpa o ambiente do IBM Spectrum Protect.
"dsmUpdateFS" na página 147	Atualiza um espaço no arquivo no armazenamento.
"dsmUpdateObj" na página 148	Atualiza as informações de objInfo associadas a um objeto de backup ativo que já se encontra no servidor ou atualizar objetos arquivados.
"dsmUpdateObjEx" na página 149	Atualiza as informações de objInfo associadas a um objeto de archive específico mesmo quando há vários objetos com o mesmo nome ou atualiza objetos de backup ativos.

Referências relacionadas:

Códigos de Retorno da API

dsmBeginGetData

A chamada de função **dsmBeginGetData** inicia uma operação de restauração ou recuperação em uma lista de objetos no armazenamento. Essa lista de objetos está contida na estrutura **dsmGetList**. O aplicativo cria essa lista com valores da consulta que antecedeu uma chamada a **dsmBeginGetData**.

O responsável pela chamada deve primeiro utilizar os campos de ordem de restauração obtidos da consulta do objeto para classificar a lista contida nessa chamada. Isso assegura que os objetos sejam restaurados do armazenamento da forma mais eficiente possível sem voltar nem remontar fitas de dados.

Ao obter objetos completos, o *dsmGetList.numObjID* máximo é DSM_MAX_GET_OBJ. Ao obter objetos parciais, o máximo é DSM_MAX_PARTIAL_GET_OBJ.

Siga a chamada a **dsmBeginGetData** com uma ou mais chamadas a **dsmGetObj** para obter cada objeto da lista. Após cada objeto ser obtido ou dados adicionais do objeto não serem mais necessários, a chamada de **dsmEndGetObj** é enviada.

Quando todos os objetos são obtidos ou **dsmEndGetObj** é cancelado, a chamada de **dsmEndGetData** é enviada. O ciclo poderá, então, ser reiniciado.

Sintaxe

```
dsInt16_t dsmBeginGetData  (dsUint32_t       dsmHandle,
   dsBool_t              mountWait,
   dsmGetType            getType,
   dsmGetList  *dsmGetObjListP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsBool_t mountWait (I)
Um valor verdadeiro ou falso booleano indica se o aplicativo cliente está disposto ou não a aguardar pela mídia off-line a ser montada se os dados necessários estiverem atualmente off-line. Se mountWait for verdadeiro, o aplicativo espera o servidor montar a mídia necessária. O aplicativo espera até que a mídia esteja montada ou que a solicitação seja cancelada.

dsmGetType getType (I)
Um tipo enumerado que consiste em **gtBackup** e **gtArchive** que indica o tipo de objeto a ser obtido.

dsmGetList *dsmGetObjListP (I)
A estrutura que contém informações sobre os objetos ou objetos parciais a serem restaurados ou recuperados. A estrutura aponta para uma lista de IDs de objetos e, no caso de uma restauração ou recuperação parcial do objeto, para uma lista de deslocamentos e comprimentos associados. Se seu aplicativo utiliza a função de restauração ou recuperação parcial do objeto, configure o campo **dsmGetList.stVersion** para **dsmGetListPORVersion**. Em uma restauração ou recuperação parcial do objeto, não é possível compactar dados ao enviá-los. Para aplicar isso, configure **ObjAttr.objCompressed** como *bTrue*.

Consulte a Figura 19 na página 74 e o Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163 para obter informações adicionais sobre essa estrutura.

Consulte "Restauração ou Recuperação Parcial de Objeto" na página 68 para obter informações adicionais sobre a restauração ou recuperação parcial do objeto.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 20. Códigos de Retorno para dsmBeginGetData

Código de retorno	Explicação
DSM_RC_ABORT_INVALID_OFFSET (33)	O deslocamento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto.
DSM_RC_ABORT_INVALID_LENGTH (34)	O comprimento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto ou o deslocamento mais o comprimento se estende além do final do objeto.
DSM_RC_NO_MEMORY (102)	Não há nenhuma RAM remanescente para concluir o pedido.
DSM_RC_NUMOBJ_EXCEED (2029)	**dsmGetList.numObjId** é maior que DSM_MAX_GET_OBJ.
DSM_RC_OBJID_NOTFOUND (2063)	O ID do objeto não foi localizado. O objeto não foi restaurado.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão de API do aplicativo cliente é diferente da versão da biblioteca do IBM Spectrum Protect.

dsmBeginQuery

A chamada de função **dsmBeginQuery** inicia um pedido de consulta ao servidor para obter informações sobre dados, espaços no arquivo e classes de gerenciamento.

Especificamente, o **dsmBeginQuery** pode consultar:
- Dados arquivados
- Dados com backup
- Dados com backup ativados
- Áreas de Arquivos
- Classes de Gerenciamento

Os dados de consulta retornados da chamada são obtidos por uma ou mais chamadas a **dsmGetNextQObj**. Quando a consulta é concluída, a chamada de **dsmEndQuery** é enviada.

Sintaxe

```
dsInt16_t dsmBeginQuery  (dsUint32_t       dsmHandle,
   dsmQueryType          queryType,
   dsmQueryBuff  *queryBuffer);
```

Parâmetros

dsUint32_t dsmHandle (I)

O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmQueryType queryType (I)

Identifica o tipo de consulta a ser executada. Designe uma das opções a seguir:

qtArchive

Consulta objetos arquivados.

qtBackup

Consulta objetos com backup.

qtBackupActive

Consulta objetos com backup ativos somente para todo o nome do espaço no arquivo transmitido. Esta consulta é chamada de um "atalho" e é uma forma eficiente de consultar objetos ativos no armazenamento.

Pré-requisito: Você deve ter efetuado logon como um usuário raiz em um sistema operacional UNIX ou Linux.

qtFilespace

Consulta espaços no arquivo registrados.

qtMC

Consulta classes de gerenciamento definidas.

qtBackupGroups

Consulta grupos fechados.

qtOpenGroups

Consulta grupos abertos.

qtProxyNodeAuth

Consulta nós para os quais esse nó pode efetuar proxy.

qtProxyNodePeer

Consulta nós do mesmo nível com o mesmo destino.

dsmQueryBuff *queryBuffer (I)

Identifica um ponteiro para um buffer mapeado para uma estrutura de dados específica. Essa estrutura está associada ao tipo de consulta transmitido. Essas estruturas contêm os critérios de seleção para cada tipo de consulta. Conclua os campos em cada estrutura para especificar o escopo para a consulta que deseja executar. O campo `stVersion` em cada estrutura contém o número de versão da estrutura.

As estruturas de dados e seus campos relacionados incluem os itens a seguir:

qryArchiveData

objName

O nome de objeto completo. É possível usar um caractere curinga como asterisco (*) ou um ponto de interrogação (?), na porção de alto ou baixo nível do nome. Um asterisco corresponde a zero ou mais caracteres, e um ponto de interrogação corresponde a um caractere. O campo `objType` de `objName` pode ter um dos valores a seguir:

- `DSM_OBJ_FILE`
- `DSM_OBJ_DIRECTORY`
- `DSM_OBJ_ANY_TYPE`

Para obter mais informações sobre nomes de alto e baixo nível, consulte o tópico a seguir: "Nomes de Nível Superior e de Nível Inferior" na página 24.

proprietário

O nome do proprietário do objeto.

insDateLowerBound

O limite inferior para a data de inserção em que o objeto foi arquivado. Para obter o limite inferior padrão, configure o componente de ano para `DATE_MINUS_INFINITE`.

insDateUpperBound

O limite superior para a data de inserção em que o objeto foi arquivado. Para obter o limite superior padrão, configure o componente de ano para `DATE_PLUS_INFINITE`.

expDateLowerBound

O limite inferior para a data de expiração. Os valores padrão para ambos campos de data de expiração são os mesmos que para os campos de data de inserção.

expDateUpperBound

O limite superior para a data de expiração.

descr

A descrição do archive. Insira um asterisco (*) para procurar todas as descrições.

qryBackupData

objName

O nome de objeto completo. É possível usar um caractere curinga como asterisco (*) ou um ponto de interrogação (?), na porção de alto ou baixo nível do nome. Um asterisco corresponde a zero ou mais caracteres, e um ponto de interrogação corresponde a um caractere. O campo `objType` de `objName` pode ter um dos valores a seguir:

- `DSM_OBJ_FILE`
- `DSM_OBJ_DIRECTORY`
- `DSM_OBJ_ANY_TYPE`

Para obter mais informações sobre nomes de alto e baixo nível, consulte o tópico a seguir: “Nomes de Nível Superior e de Nível Inferior” na página 24.

proprietário

O nome do proprietário do objeto.

objState

É possível consultar um dos estados de objeto a seguir:

- `DSM_ACTIVE`
- `DSM_INACTIVE`
- `DSM_ANY_MATCH`

pitDate

O valor de um momento específico. Uma consulta com este campo retorna o objeto mais recente do qual foi feito backup depois desta data e hora. O `objState` pode estar ativo ou inativo. Objetos que foram excluídos antes do `pitDate` naõ são retornados. Por exemplo:

```
Mon - backup ABC(1), DEF, GHI
Tue - backup ABC(2), delete DEF
Thr - backup ABC(3)
```

Na sexta-feira, chame a consulta com um valor de momento de quarta-feira às 12:00:00 a.m. A chamada retorna as informações a seguir:

```
ABC(2) - an Inactive copy
GHI    - an Active copy
```

A chamada não retorna DEF porque este objeto foi excluído antes do valor do momento.

qryABackupData

objName

O nome de objeto completo. É possível usar um caractere curinga como asterisco (*) ou um ponto de interrogação (?), na porção de alto ou baixo nível do nome. Um asterisco corresponde a zero ou mais caracteres, e um ponto de interrogação corresponde a um caractere. O campo objType de objName pode ter um dos valores a seguir:

- DSM_OBJ_FILE
- DSM_OBJ_DIRECTORY
- DSM_OBJ_ANY_TYPE

Para obter mais informações sobre nomes de alto e baixo nível, consulte o tópico a seguir: "Nomes de Nível Superior e de Nível Inferior" na página 24.

qryFSData

fsName

Digite o nome de um espaço no arquivo específico nesse campo ou digite um asterisco (*) para recuperar informações sobre todos os espaços no arquivo registrados.

qryMCData

mcName

Digite o nome de uma classe de gerenciamento específica ou digite uma cadeia vazia (" ") para recuperar informações sobre todas as classes de gerenciamento.

Nota: Você não pode utilizar um asterisco (*).

mcDetail

Determine se as informações nos grupos de cópia do backup e archive da classe de gerenciamento são retornadas. Os valores a seguir são válidos:

- bTrue
- bFalse

qryBackupGroup:

groupType

O tipo de grupo é DSM_GROUPTYPE_PEER.

fsName

O nome do espaço no arquivo.

owner

O ID do proprietário.

groupLeaderObjId

O ID do objeto do líder do grupo.

objType
O tipo de objeto.

qryProxyNodeAuth:

targetNodeName

O nome do nó de destino.

peerNodeName

O nome do nó do peer.

hlAddress

O endereço do peer do nome de alto nível.

llAddress

O endereço do peer do nome de baixo nível.

qryProxyNodePeer:

targetNodeName

O nome do nó de destino.

peerNodeName

O nome do nó do peer.

hlAddress

O endereço do peer do nome de alto nível.

llAddress

O endereço do peer do nome de baixo nível.

Códigos de retorno

A tabela a seguir descreve os códigos de retorno para a chamada de função **dsmBeginQuery**.

Tabela 21. Códigos de Retorno para dsmBeginQuery

Código de retorno	Número do código de retorno	Explicação
DSM_RC_NO_MEMORY	102	Não há memória suficiente para concluir a solicitação.
DSM_RC_FILE_SPACE_NOT_FOUND	124	O espaço no arquivo especificado não foi localizado.
DSM_RC_NO_POLICY_BLK	2007	As informações de política do servidor não estavam disponíveis.
DSM_RC_INVALID_OBJTYPE	2010	O tipo de objeto é inválido.
DSM_RC_INVALID_OBJOWNER	2019	O nome do proprietário do objeto é inválido.
DSM_RC_INVALID_OBJSTATE	2024	A condição do objeto é inválida.

Tabela 21. Códigos de Retorno para dsmBeginQuery (continuação)

Código de retorno	Número do código de retorno	Explicação
DSM_RC_WRONG_VERSION_PARM	2065	A versão de API do aplicativo cliente é diferente da versão da biblioteca do IBM Spectrum Protect.

dsmBeginTxn

A chamada de função **dsmBeginTxn** inicia uma ou mais transações do IBM Spectrum Protect que iniciam uma ação completa; todas as ações são bem-sucedidas ou nenhuma é bem-sucedida. Uma ação pode ser uma única chamada ou uma série de chamadas. Por exemplo, uma chamada de **dsmSendObj** seguida por várias chamadas de **dsmSendData** pode ser considerada uma única ação. De forma semelhante, uma chamada de **dsmSendObj** com **dataBlkPtr** que indica uma área de dados contendo o objeto do qual deve ser feito backup também é considerada uma única ação.

Tente agrupar mais de um objeto em uma única transação para operações de transferência de dados. O agrupamento de objetos resulta em aprimoramentos significativos no sistema IBM Spectrum Protect. De uma perspectiva do cliente e do servidor, uma determinada quantia de desgaste geral incorre ao iniciar e encerrar cada transação.

Há limites para o que pode ser executado em uma única transação. Essas restrições incluem:

- Um número máximo de objetos que podem ser enviados ou excluídos em uma única transação. Esse limite está localizado nos dados retornados por **dsmQuerySessInfo** no campo *ApiSessInfo.maxObjPerTxn*. Isso corresponde à opção *TxnGroupMax* do servidor.
- Todos os objetos enviados ao servidor (backup ou archive) em uma única transação devem ter o mesmo destino de cópia definido na ligação da classe de gerenciamento para o objeto. Esse valor está localizado nos dados que **dsmBindMC** retorna nos campos **mcBindKey.backup_copy_dest** ou **mcBindKey.archive_copy_dest**.

Com a API, o cliente aplicativo pode monitorar e controlar essas restrições ou a API pode monitorar essas restrições. Se a API estiver monitorando as restrições, códigos de retorno apropriados das chamadas de API informam o cliente aplicativo quando uma ou mais restrições forem atingidas.

Sempre corresponda uma chamada de **dsmBeginTxn** a uma chamada de **dsmEndTxn** para otimizar o conjunto de ações em um par de chamadas de **dsmBeginTxn** e **dsmEndTxn**.

Sintaxe

```
dsInt16_t dsmBeginTxn  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 22. Códigos de Retorno para dsmBeginTxn

Código de retorno	Explicação
DSM_RC_ABORT_NODE_NOT_AUTHORIZED (36)	FROMNODE nem FROMOWNER não são permitidos para operações TXN.

dsmBindMC

Uma chamada de função **dsmBindMC** associa ou liga uma classe de gerenciamento ao objeto transmitido. O objeto é passado por meio da lista Incluir-Excluir apontada para o arquivo de opções. Se uma correspondência não for localizada na lista Incluir para uma classe de gerenciamento específica, a classe de gerenciamento padrão é designada. A lista Excluir pode impedir backup de objetos, mas não archive.

O cliente aplicativo pode utilizar os parâmetros retornados na estrutura `mcBindKey` para determinar se deve ser feito backup ou archive desse objeto ou se uma nova transação deve ser iniciada, devido a diferentes destinos de cópia. Consulte **dsmBeginTxn** para obter mais informações.

Chame **dsmBindMC** antes de chamar **dsmSendObj**, pois cada objeto deve ter uma classe de gerenciamento associada a ela. Essa chamada deve ser executada em uma transação ou fora de uma transação. Por exemplo, dentro de uma transação com vários objetos, se **dsmBindMC** indica que o objeto tem um destino de cópia diferente do que o objeto anterior, a transação deve ser encerrada e uma nova transação iniciada. Nesse caso, não é requerida outra **dsmBindMC**, pois uma já foi executada para o objeto.

Sintaxe

```
dsInt16_t dsmBindMC  (dsUint32_t       dsmHandle,
   dsmObjName             *objNameP,
   dsmSendType            sendType,
   mcBindKey   *mcBindKeyP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmObjName *objNameP (I)
Um ponteiro para a estrutura que contém o nome do espaço no arquivo, o nome do objeto de nível superior, o nome do objeto de nível inferior e o tipo de objeto.

dsmSendType sendType (I)
Identifica se essa ligação da classe de gerenciamento é executada para envios de archive ou de backup. Os possíveis valores para essa chamada incluem:

Nome	Descrição
stBackup	Um objeto de backup
stArchive	Um objeto de archive
stBackupMountWait	Um objeto de backup
stArchiveMountWait	Um objeto de archive

Para a chamada de **dsmBindMC**, stBackup e stBackupMountWait são equivalentes e stArchive e stArchiveMountWait também.

mcBindKey *mcBindKeyP (O)
Esse é o endereço de uma estrutura mcBindKey onde as informações da classe de gerenciamento são retornadas. O cliente aplicativo pode utilizar as informações retornadas aqui para determinar se esse objeto se encaixa em uma transação com vários objetos ou para executar uma consulta de classe de gerenciamento na classe de gerenciamento ligada ao objeto.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 23. Códigos de Retorno para dsmBindMC

Código de retorno	Explicação
DSM_RC_NO_MEMORY (102)	Não há nenhuma RAM remanescente para concluir o pedido.
DSM_RC_INVALID_PARM (109)	Um dos parâmetros transmitidos tem um valor inválido.
DSM_RC_TL_EXCLUDED (185)	O objeto de backup é excluído e não pode ser enviado.
DSM_RC_INVALID_OBJTYPE (2010)	O tipo de objeto é inválido.
DSM_RC_INVALID_SENDTYPE (2022)	Tipo de envio inválido.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.

dsmChangePW

A chamada de função **dsmChangePW** muda uma senha de IBM Spectrum Protect. Em um sistema operacional com vários usuários, como UNIX ou Linux, somente o usuário raiz ou o usuário autorizado podem utilizar essa chamada.

Em sistemas operacionais Windows, é possível especificar a senha no arquivo dsm.opt. Nessa situação, **dsmChangePW** não atualiza o arquivo dsm.opt. Depois que a chamada a **dsmChangePW** for realizada, você deverá atualizar o arquivo dsm.opt separadamente.

Essa chamada deve ser processada com sucesso se **dsmInitEx** retornar DSM_RC_VERIFIER_EXPIRED. A sessão será encerrada se a chamada de **dsmChangePW** falhar nessa situação.

Se **dsmChangePW** for chamado por alguma outra razão, a sessão permanecerá aberta, independentemente do código de retorno.

Sintaxe

```
dsInt16_t dsmChangePW  (dsUint32_t       dsmHandle,
   char                  *oldPW,
   char    *newPW);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

char *oldPW (I)
A senha antiga do responsável pela chamada. O comprimento máximo é DSM_MAX_VERIFIER_LENGTH.

char *newPW (I)
A nova senha do responsável pela chamada. O comprimento máximo é DSM_MAX_VERIFIER_LENGTH.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 24. Códigos de Retorno para dsmChangePW

Código de retorno	Explicação
DSM_RC_ABORT_BAD_VERIFIER (6)	Uma senha incorreta foi inserida.
DSM_RC_AUTH_FAILURE (137)	Falha de autenticação. A senha antiga está incorreta.
DSM_RC_NEWPW_REQD (2030)	Um valor deve ser digitado para a nova senha.
DSM_RC_OLDPW_REQD (2031)	Um valor deve ser digitado para a senha antiga.
DSM_RC_PASSWD_TOOLONG (2103)	A senha especificada é muito longa.
DSM_RC_NEED_ROOT (2300)	O responsável pela chamada da API deve ser um usuário raiz ou um usuário autorizado.

dsmCleanUp

A chamada de função **dsmCleanUp** é utilizada se **dsmSetUp** tiver sido chamada. A chamada de função **dsmCleanUp** deve ser chamada após **dsmTerminate**. Não será possível fazer nenhuma outra chamada após a chamada de **dsmCleanUp**.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t DSMLINKAGE dsmCleanUp
    (dsBool_t        mtFlag);
```

Parâmetros

dsBool_t mtFlag (I)
Esse parâmetro especifica que a API foi utilizada no modo de encadeamento único ou multiencadeado. Os valores possíveis são:

- DSM_SINGLETHREAD
- DSM_MULTITHREAD

dsmDeleteAccess

A chamada de função **dsmDeleteAccess** exclui as regras de autorização atuais para versões de backup ou cópias arquivadas de seus objetos. Ao excluir uma regra de autorização, você revoga o acesso que um usuário tem aos arquivos especificados pela regra.

Ao utilizar **dsmDeleteAccess**, é possível excluir somente uma regra por vez. Obtenha o ID da regra através do comando **dsmQueryAccess**.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t DSMLINKAGE dsmDeleteAccess
             (dsUint32_t          dsmHandle,
              dsUint32_t          ruleNum) ;
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsUint32_t ruleNum (I)
O ID da regra para a regra de acesso excluída. Esse valor é obtido de uma chamada de função **dsmQueryAccess**.

dsmDeleteFS

A chamada de função **dsmDeleteFS** exclui um espaço no arquivo do armazenamento. Para excluir um espaço no arquivo, você deve ter as permissões apropriadas fornecidas pelo administrador do IBM Spectrum Protect. Para determinar se você tem as permissões necessárias, chame **dsmQuerySessInfo**. Essa chamada de função retorna uma estrutura de dados do tipo *ApiSessInfo*, que inclui dois campos, *archDel* e *backDel*.

Nota:

- Em um sistema operacional UNIX ou Linux, somente um usuário raiz ou um usuário autorizado pode excluir um espaço no arquivo.
- Se o espaço no arquivo que precisa ser excluído contiver versões de backup, você deve ter autoridade de exclusão de backup (***backDel*** = BACKDEL_YES). Se o espaço no arquivo contiver cópias de archive, você deverá ter autoridade de exclusão de archive (*archDel* = ARCHDEL_YES). Se o espaço no arquivo contiver ambas, versões de backup e cópias de archive, você deve ter ambos tipos de autoridade de exclusão.
- Ao utilizar um servidor de gerenciador de archive, um espaço no arquivo não poderá realmente ser removido. Essa chamada de função retornará *rc=0*, apesar de o espaço no arquivo não ter sido realmente excluído. A única maneira de verificar se o espaço no arquivo foi excluído é emitindo uma consulta de espaço no arquivo no servidor.
- A função de exclusão do espaço no arquivo do servidor IBM Spectrum Protect é um processo de segundo plano. Caso ocorram erros além dos detectados antes da transmissão de um código de retorno, eles serão registrados no log do servidor IBM Spectrum Protect.

Sintaxe

```
dsInt16_t dsmDeleteFS  (dsUint32_t       dsmHandle,
   char                  *fsName,
   unsigned char   repository);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

char *fsName (I)
Um ponteiro para o nome do espaço no arquivo a ser excluído. O caractere curinga não é permitido.

unsigned char repository (I)
Indica se o espaço no arquivo a ser excluído é um repositório de backup, um repositório de archive ou ambos. Os possíveis valores para esse campo incluem:

```
DSM_ARCHIVE_REP     /* repositório de archive   */
DSM_BACKUP_REP      /* repositório de backup    */
DSM_REPOS_ALL       /* todos os tipos de repositório */
```

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 25. Códigos de Retorno para dsmDeleteFS

Código de retorno	Explicação
DSM_RC_ABORT_NOT_AUTHORIZED (27)	Você não tem a autoridade necessária para excluir o espaço no arquivo.
DSM_RC_INVALID_REPOS (2015)	Valor inválido para o repositório.
DSM_RC_FSNAME_NOTFOUND (2060)	Nome do espaço no arquivo não localizado.
DSM_RC_NEED_ROOT (2300)	O responsável pela chamada da API deve ser um usuário root.

dsmDeleteObj

A chamada de função **dsmDeleteObj** chama objetos de backup inativos, exclui objetos de backup ou exclui objetos de archive do armazenamento. O tipo **dtBackup** deixa inativa somente a cópia de backup atualmente ativa. O tipo **dtBackupID** remove do servidor o ID de objeto especificado. Chame essa função a partir de uma transação.

Consulte **dsmBeginTxn** para obter mais informações.

Antes de enviar **dsmDeleteObj**, envie a seqüência de consulta descrita em "Consultando o sistema IBM Spectrum Protect" na página 34 para obter as informações para **delInfo**. A chamada a **dsmGetNextQObj** retorna uma estrutura de dados denominada **qryRespBackupData** para consultas de backup ou **qryRespArchiveData** para consultas de archive. Essas estruturas de dados contêm as informações necessárias para **delInfo**.

O valor de **maxObjPerTxn** determina o número máximo de objetos que pode ser excluído em uma única transação. Para obter esse valor, chame **dsmQuerySessInfo**.

Dica: Seu nó deve ter a permissão adequada configurada pelo administrador. Para excluir objetos de archive, você deve ter autoridade de exclusão de archive. Não é necessário ter autoridade de exclusão de backup para tornar inativo um objeto de backup.

Sintaxe

```
dsInt16_t dsmDeleteObj  (dsUint32_t       dsmHandle,
   dsmDelType            delType,
   dsmDelInfo  delInfo)
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmDelType delType (I)
Indica o tipo de objeto (backup or archive) a ser excluído. Os valores possíveis são:

Nome	Descrição
dtArchive	O objeto a ser excluído foi arquivado anteriormente.
dtBackup	Foi feito backup anteriormente do objeto a ser definido como inativo.
dtBackupID	Foi feito backup anteriormente do objeto a ser excluído. **Restrição:** A utilização de **delType** com *objID* remove o objeto de backup do servidor. Somente o proprietário de um objeto pode excluí-lo. É possível excluir qualquer versão (ativa ou inativa) de um objeto. O servidor reconcilia as versões. Se você excluir uma versão ativa de um objeto, a primeira versão inativa torna-se ativa. Se você excluir uma versão inativa de um objeto, todas as versões mais antigas serão avançadas. O nó deve ser registrado com a permissão **backDel**.

dsmDelInfo delInfo (I)
Uma estrutura cujos campos identificam o objeto. Os campos são diferentes, dependendo de se o objeto é um objetos de backup ou um objeto de archive. A estrutura para tonar um objeto de backup inativo, `delBack`, contém o nome do objeto e o grupo de cópias do objeto. A estrutura de um objeto de archive, `delArch`, contém o ID do objeto.

A estrutura para remover um objeto de backup, `delBackID`, contém o ID do objeto.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 26. Códigos de Retorno para dsmDeleteObj

Código de retorno	Explicação
DSM_RC_FS_NOT_REGISTERED (2061)	O nome do espaço no arquivo não está registrado.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.

dsmEndGetData

A chamada de função **dsmEndGetData** termina uma sessão de **dsmBeginGetData** que obtém objetos do armazenamento.

A chamada de função **dsmEndGetData** é iniciada após todos os objetos que você deseja restaurar terem sido processados ou termina o processo de obtenção prematuramente. Chame **dsmEndGetData** para encerrar uma sessão de **dsmBeginGetData** antes de poder continuar outros processamentos.

Dependendo de quando **dsmEndGetData** é chamada, a API pode precisar encerrar o processamento parcial de um fluxo de dados antes de o processo poder ser parado. O responsável pela chamada, portanto, não deve esperar um retorno imediato dessa chamada. Utilize **dsmTerminate** se o aplicativo precisar fechar a sessão e encerrar a restauração imediatamente.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t dsmEndGetData  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmEndGetDataEx

A chamada de função **dsmEndGetDataEx** fornece o total de bytes sem a LAN que foram enviados. É uma extensão da chamada de função **dsmEndGetData**.

Sintaxe

Não há códigos de retorno específicos para essa chamada.

```
dsInt16_t dsmEndGetDataEx (dsmEndGetDataExIn_t * dsmEndGetDataExInP,
                           dsmEndGetDataExOut_t * dsmEndGetDataExOutP);
```

Parâmetros

dsmEndGetDataExIn_t *dsmEndGetDataExInP (I)
Transmite o objeto end get, dsmHandle, que identifica a sessão e a associa às chamadas subseqüentes.

dsmEndGetDataExOut_t * dsmEndGetDataExOutP (O)
Essa estrutura contém este parâmetro de entrada:

totalLFBytesRecv
O total de bytes sem a LAN que são recebidos.

dsmEndGetObj

A chamada de função **dsmEndGetObj** termina uma sessão de **dsmGetObj** que obtém dados para um objeto especificado.

Inicie a chamada de **dsmEndGetObj** após um final de dados ser recebido para o objeto. Isso indica que todos os dados foram recebidos ou que nenhum outro dado será recebido para o objeto. Antes de poder iniciar outra chamada de **dsmGetObj**, você deve chamar **dsmEndGetObj**.

Dependendo de quando **dsmEndGetObj** é chamada, a API pode precisar encerrar o processamento parcial de um fluxo de dados antes de o processo poder parar. Não espere um retorno imediato dessa chamada.

Sintaxe

```
dsInt16_t dsmEndGetObj  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 27. Códigos de Retorno para dsmEndGetObj

Código de retorno	Explicação
DSM_RC_NO_MEMORY (102)	Não há nenhuma RAM remanescente para concluir o pedido.

dsmEndQuery

A chamada de função **dsmEndQuery** significa o final de uma ação **dsmBeginQuery**. O cliente aplicativo envia **dsmEndQuery** para concluir uma consulta. Essa chamada é enviada após todas as respostas de consulta serem obtidas através do **dsmGetNextQObj** ou é enviada para encerrar uma consulta antes de todos os dados serem retornados.

Dica: O IBM Spectrum Protect continua a enviar os dados de consulta do servidor para o cliente nesse caso, mas a API descarta quaisquer dados restantes.

Após o envio de **dsmBeginQuery**, **dsmEndQuery** deve ser enviada antes do início de qualquer outra atividade.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t dsmEndQuery  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmEndSendObj

A chamada de função **dsmEndSendObj** indica o final dos dados enviados para armazenamento.

Digite a chamada de função **dsmEndSendObj** para indicar o final de dados das chamadas de **dsmSendObj** e de **dsmSendData**. Ocorre uma violação de protocolo se isso não for realizado. A exceção a essa regra é se você chamar **dsmEndTxn** para encerrar a transação. Fazer isso descarta todos os dados enviados para a transação.

Sintaxe

```
dsInt16_t dsmEndSendObj  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 28. Códigos de Retorno para dsmEndSendObj

Código de retorno	Explicação
DSM_RC_NO_MEMORY (102)	Não há nenhuma RAM remanescente para concluir o pedido.

dsmEndSendObjEx

A chamada de função **dsmEndSendObjEx** fornece informações adicionais referentes ao número de bytes processados. As informações incluem: total de bytes enviados, informações de compactação, bytes sem a LAN e informações de deduplicação.

A chamada de função **dsmEndSendObjEx** é uma extensão da chamada de função **dsmEndSendObj**.

Sintaxe

```
dsInt16_t dsmEndSendObjEx (dsmEndSendObjExIn_t  *dsmEndSendObjExInP,
                           dsmEndSendObjExOut_t *dsmEndSendObjExOutP);
```

Parâmetros

dsmEndSendObjExIn_t *dsmEndSendObjExInP (I)
Esse parâmetro transmite o objeto end send, dsmHandle, que identifica a sessão e a associa às chamadas subseqüentes.

dsmEndSendObjExOut_t *dsmEndSendObjExOutP (O)
Este parâmetro transmite as informações de término do objeto send:

Nome	Descrição
totalBytesSent	O número total de bytes lidos do aplicativo.
objCompressed	Um sinalizador que é exibido se o objeto tiver sido compactado.
totalCompressedSize	O tamanho total em bytes após a compactação.
totalLFBytesSent	O total de bytes sem a LAN que foram enviados.
objDeduplicated	Um sinalizador que é exibido se o objeto tiver sido deduplicado pela API.
totalDedupSize	Total de bytes enviados após a deduplicação.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 29. Códigos de Retorno para dsmEndSendObjEx

Código de retorno	Explicação
DSM_RC_NO_MEMORY (102)	Não há nenhuma RAM remanescente para concluir o pedido.

dsmEndTxn

A chamada de função **dsmEndTxn** termina uma transação do IBM Spectrum Protect. Faça um para da chamada de função **dsmEndTxn** com **dsmBeginTxn** para identificar a chamada ou conjunto de chamadas consideradas uma transação. O aplicativo cliente pode especificar na chamada **dsmEndTxn** se a transação deve ser confirmada ou terminada.

Execute todas as chamadas a seguir dentro dos limites de uma transação:

- **dsmSendObj**
- **dsmSendData**
- **dsmEndSendObj**
- **dsmDeleteObj**

Sintaxe

```
dsInt16_t dsmEndTxn  (dsUint32_t       dsmHandle,
   dsUint8_t             vote,
   dsUint16_t  *reason);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsUint8_t vote (I)
Indica se o aplicativo cliente confirma todas as ações que são feitas entre a chamada **dsmBeginTxn** anterior e esta chamada. Os valores a seguir são possíveis:

```
DSM_VOTE_COMMIT     /* confirmar transação atual    */
DSM_VOTE_ABORT      /* recuperar transação atual */
```

Use DSM_VOTE_ABORT apenas se seu aplicativo localizar um motivo para parar a transação.

dsUint16_t *reason (O)
Se a chamada para **dsmEndTxn** terminar com um erro, ou o valor de vote não for aceito, este parâmetro possui um código de razão que indica porque o voto falhou. O código de retorno da chamada pode ser zero e o código de razão pode ser diferente de zero. Portanto, o cliente aplicativo deve sempre verificar erros no código de retorno e na razão (if (rc || reason)) antes que você possa supor uma conclusão bem-sucedida.

Se o aplicativo especificar um voto de DSM_VOTE_ABORT, o código de razão é DSM_RS_ABORT_BY_CLIENT (3). Consulte o Apêndice A, "Arquivo de Origem dos Códigos de Retorno do API : dsmrc.h", na página 153 para obter uma lista de possíveis códigos de razão. Os números 1 a 50 na lista de códigos de retorno estão reservados para os códigos de razão. Se o servidor terminar a transação, o código de retorno é DSM_RC_CHECK_REASON_CODE. Nesse caso, o valor da razão conterá informações adicionais sobre a causa da interrupção.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 30. Códigos de Retorno para dsmEndTxn

Código de retorno	Explicação
DSM_RC_ABORT_CRC_FAILED (236)	O CRC recebido do servidor não corresponde ao CRC que foi calculado pelo cliente.
DSM_RC_INVALID_VOTE (2011)	O valor especificado para vote não é válido.
DSM_RC_CHECK_REASON_CODE (2302)	A transação foi interrompida. Verifique o campo da razão.
DSM_RC_ABORT_STGPOOL_COPY_CONT_NO (241)	A gravação em um dos conjuntos de armazenamentos de cópia falhou e a opção do conjunto de armazenamentos COPYCONTINUE do IBM Spectrum Protect está configurada como NO. A transação é encerrada.
DSM_RC_ABORT_RETRY_SINGLE_TXN (242)	Esse código de interrupção indica que a transação atual foi interrompida devido a um problema durante uma operação de armazenamento. O problema pode ser resolvido enviando cada arquivo em uma transação individual. Este erro é comum nas circunstâncias a seguir: • O próximo conjunto de armazenamentos possui uma lista de conjunto de armazenamento de cópia diferente. • A operação é alternada para este conjunto no meio de uma transação. .

dsmEndTxnEx

A chamada de função **dsmEndTxnEx** fornece informações do ID do objeto líder do grupo para utilização com a chamada de função **dsmGroupHandler**. É uma extensão da chamada de função **dsmEndTxn**.

Sintaxe

```
dsInt16_t dsmEndTxnEx (dsmEndTxnExIn_t *dsmEndTxnExInP
                       dsmEndTxnExOut_t *dsmEndTxnExOutP);
```

Parâmetros

dsmEndTxnExIn_t *dsmEndTxnExInP (I)
Esta estrutura contém os seguintes parâmetros:

dsmHandle
O identificador que identifica a sessão e a associa a chamadas subsequentes da IBM Spectrum Protect.

dsUint8_t vote (I)
Indica se o aplicativo cliente deseja ou não confirmar todas as ações realizadas entre a chamada de **dsmBeginTxn** anterior e esta chamada. Os valores possíveis são:

```
DSM_VOTE_COMMIT     /* confirmar transação atual    */
DSM_VOTE_ABORT      /* recuperar transação atual */
```

Utilize DSM_VOTE_ABORT somente se seu aplicativo tiver encontrado uma razão para parar a transação.

dsmEndTxnExOut_t *dsmEndTxnExOutP (O)
Esta estrutura contém os seguintes parâmetros:

dsUint16_t *reason (O)
Se a chamada para **dsmEndTxnEx** terminar com um erro ou se o valor de *vote* não for aceito, esse parâmetro terá um código de razão que indica a razão da falha de vote.

Dica: O código de retorno da chamada pode ser zero e o código de razão pode ser diferente de zero. Portanto, o cliente aplicativo deve sempre verificar erros no código de retorno e na razão (`if (rc || reason)`) antes que você possa supor uma conclusão bem-sucedida.

Se o aplicativo especificar um vote de DSM_VOTE_ABORT, o código de razão será DSM_RS_ABORT_BY_CLIENT (3). Consulte o Apêndice A, "Arquivo de Origem dos Códigos de Retorno do API : `dsmrc.h`", na página 153 para obter uma lista de possíveis códigos de razão. Os números 1 a 50 na lista de códigos de retorno estão reservados para os códigos de razão. Se o servidor encerrar a transação, o código de retorno será DSM_RC_CHECK_REASON_CODE. Nesse caso, o valor da razão conterá informações adicionais sobre a causa da interrupção.

groupLeaderObjId
O ID do objeto líder do grupo retornado quando o sinalizador DSM_ACTION_OPEN é utilizado com a chamada de **dsmGroupHandler**.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 31. Códigos de Retorno para dsmEndTxnEx

Código de retorno	Explicação
DSM_RC_INVALID_VOTE (2011)	O valor especificado para vote é inválido.
DSM_RC_CHECK_REASON_CODE (2302)	A transação foi interrompida. Verifique o campo da razão.
DSM_RC_ABORT_STGPOOL_COPY_CONT_NO (241)	A gravação para um dos conjuntos de armazenamento de cópias falhou e a opção COPYCONTINUE do conjunto de armazenamento do IBM Spectrum Protect foi configurada para NO. A transação é encerrada.
DSM_RC_ABORT_RETRY_SINGLE_TXN (242)	Durante a operação de gravação simultânea, um objeto na transação está indo para um destino com conjuntos de armazenamentos de cópia diferentes. Encerre a transação atual e envie cada objeto novamente em sua própria transação.

dsmGetData

A chamada de função **dsmGetData** obtém um fluxo de bytes de dados a partir de IBM Spectrum Protect e o coloca no buffer do responsável pela chamada. O cliente aplicativo chama **dsmGetData** quando há mais dados a receber de uma chamada anterior de **dsmGetObj** ou de **dsmGetData**.

Sintaxe

```
dsInt16_t dsmGetData  (dsUint32_t       dsmHandle,
   DataBlk *dataBlkPtr);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

DataBlk *dataBlkPtr (I/O)
Aponta para uma estrutura que inclui um ponteiro para o buffer para os dados recebidos e o tamanho do buffer. No retorno, essa estrutura contém o número de bytes que é realmente transferido. Consulte o Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163 para obter a definição do tipo.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 32. Códigos de Retorno para dsmGetData

Código de retorno	Explicação
DSM_RC_ABORT_INVALID_OFFSET (33)	O deslocamento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto.
DSM_RC_ABORT_INVALID_LENGTH (34)	O comprimento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto ou o deslocamento mais o comprimento se estende além do final do objeto.
DSM_RC_FINISHED (121)	O processamento é concluído. O último buffer foi recebido. Verifique `numBytes` para obter a quantidade de dados e, em seguida, chame IBM Spectrum ProtectdsmEndGetObj.
DSM_RC_NULL_DATABLKPTR (2001)	O ponteiro do bloco de dados é nulo.
DSM_RC_ZERO_BUFLEN (2008)	O comprimento do buffer é zero para o ponteiro do bloco de dados.
DSM_RC_NULL_BUFPTR (2009)	O ponteiro do buffer é nulo para o ponteiro do bloco de dados.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_MORE_DATA (2200)	Há mais dados a serem obtidos.

dsmGetBufferData

A chamada de função **dsmGetBufferData** recebe um fluxo de bytes de dados do IBM Spectrum Protect por meio de um buffer. Após cada chamada, o aplicativo precisa copiar os dados e liberar o buffer através de uma chamada a **dsmReleaseBuffer**. Se o número de buffers mantido pelo aplicativo for igual a numTsmBuffers especificado na chamada de **dsmInitEx**, a função **dsmGetBufferData** será bloqueada até que **dsmReleaseBuffer** seja chamado.

Sintaxe

```
dsInt16_t  dsmGetBufferData   (getDatatExIn_t        *dsmGetBufferDataExInP,
                               getDataExOut_t        *dsmGetBufferDataExOutP) ;
```

Parâmetros

getDataExIn_t * dsmGetBufferDataExInP (I)
Essa estrutura contém o seguinte parâmetro de entrada.

dsUint32_t dsmHandle
O identificador que identifica a sessão e a associa a uma chamada anterior de **dsmInitEx**.

getDataExOut_t * dsmGetBufferDataExOutP (0)
Essa estrutura contém os seguintes parâmetros de saída.

dsUint8_t tsmBufferHandle(0)
O identificador que identifica o buffer recebido.

char *dataPtr(0)
O endereço no qual os dados foram gravados.

dsUint32_t numBytes(0)
O número de bytes real gravados pelo IBM Spectrum Protect.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 33. Códigos de Retorno para dsmGetBufferData

Código de retorno	Explicação
DSM_RC_BAD_CALL_SEQUENCE (2041)	A chamada não foi emitida no estado apropriado.
DSM_RC_OBJ_ENCRYPTED (2049)	Essa função não pode ser utilizada para objetos criptografados.
DSM_RC_OBJ_COMPRESSED (2048)	Essa função não pode ser utilizada para objetos compactados.
DSM_RC_BUFF_ARRAY_ERROR (2045)	Ocorreu um erro de matriz do buffer.

dsmGetNextQObj

A chamada de função **dsmGetNextQObj** obtém a próxima resposta da consulta de uma chamada **dsmBeginQuery** anterior e posiciona a resposta no buffer do responsável pela chamada.

A chamada de **dsmGetNextQObj** é chamada uma ou mais vezes. Cada vez que a função é chamada, um único registro de consulta é recuperado, ou um erro ou um código de retorno DSM_RC_FINISHED é retornado. Se DSM_RC_FINISHED for retornado, não há mais dados a serem processados. Quando todos os dados de consulta forem recuperados, ou se nenhum dado de consulta for necessário, envie a chamada **dsmEndQuery** para terminar o processo de consulta.

O parâmetro **dataBlkPtr** deve apontar para um buffer que esteja definido com o tipo de estrutura **qryResp*Data**. O contexto no qual **dsmGetNextQObj** é chamado determina o tipo de estrutura digitada na resposta da consulta.

Sintaxe

```
dsInt16_t dsmGetNextQObj (dsUint32_t   dsmHandle,
   DataBlk *dataBlkPtr);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

DataBlk *dataBlkPtr (I/O)
Aponta para uma estrutura que inclui um ponteiro para o buffer para os dados a serem recebidos e o tamanho do buffer. Este buffer é a estrutura de resposta **qryResp*Data**. No retorno, esta estrutura contém o número de bytes que é transferido. A estrutura que está associada com cada tipo de consulta é descrito na tabela a seguir. Para obter mais informações sobre a definição de tipo de **DataBlk**, consulte o tópico a seguir: Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163.

Tabela 34. Estrutura do Ponteiro DataBlk

Consultar	Estrutura da resposta	Campos de interesse especial
qtArchive	**qryRespArchiveData**	**sizeEstimate** Contém o valor que é transmitido em uma chamada **dsmSendObj** anterior. **mediaClass** Pode ter um valor de MEDIA_FIXED se o objeto estiver no disco, ou MEDIA_LIBRARY se o objeto estiver na fita. **clientDeduplicated** Indica se este objeto foi deduplicado pelo cliente.
qtBackup	**qryRespBackupData**	**restoreOrderExt** É do tipo dsUint16_t. Classifique neste campo quando vários objetos forem restaurados em uma chamada **dsmBeginGetData**. Há um exemplo de código de classificação para esta chamada na amostra da API, dapiqry.c. Para um exemplo de classificação, consulte o tópico a seguir: Figura 16 na página 70. **sizeEstimate** Contém o valor que é transmitido em uma chamada **dsmSendObj** anterior. **mediaClass** Pode ter um valor de MEDIA_FIXED se o objeto estiver no disco, ou MEDIA_LIBRARY se o objeto estiver na fita. **clientDeduplicated** Indica se este objeto foi deduplicado pelo cliente.
qtBackupActive	**qryARespBackupData**	
qtBackupGroups	**qryRespBackupData**	**dsBool_t isGroupLeader** Se true, significa que esse objeto é um líder de grupo.
qtOpenGroups	**qryRespBackupData**	**dsBool_t isOpenGroup;** Se true, significa que este grupo está aberto e não concluído.

Tabela 34. Estrutura do Ponteiro DataBlk (continuação)

Consultar	Estrutura da resposta	Campos de interesse especial
qtFilespace	**qryRespFSData**	**backStartDate** Contém o registro de data e hora do servidor quando o espaço no arquivo é atualizado com a ação **backStartDate**. **backCompleteDate** Contém o registro de data e hora do servidor quando o espaço no arquivo é atualizado com a ação **backCompleteDate**. **lastReplStartDate** Contém o registro de data e hora para a última vez que a replicação foi iniciada no servidor. **lastReplCmpltDate** Contém o registro de data e hora para a última vez que a replicação foi concluída, mesmo que tenha ocorrido uma falha. **lastBackOpDateFromServer** Contém o registro de data e hora do último armazenamento que foi salvo no servidor. **lastBackOpDateFromLocal** Contém o registro de data e hora do último armazenamento que foi salvo no cliente.
qtMC	**qryRespMCData** **qryRespMCDetailData**	
qtProxyNodeAuth	**qryRespProxyNodeData** **targetNodeName** **peerNodeName** **hlAddress** **llAddress**	
qtProxyNodePeer	**qryRespProaxyNodeData** **targetNodeName** **peerNodeName** **hlAddress** **llAddress**	

Códigos de retorno

TA tabela a seguir descreve os códigos de retorno para a chamada de função **dsmGetNextQObj**.

Tabela 35. Códigos de Retorno para a Chamada de Função **dsmGetNextQObj**

Código de retorno	Número do código de retorno	Descrição
DSM_RC_ABORT_NO_MATCH	2	Nenhuma correspondência para a consulta foi solicitada.

Tabela 35. Códigos de Retorno para a Chamada de Função **dsmGetNextQObj** *(continuação)*

Código de retorno	Número do código de retorno	Descrição
DSM_RC_FINISHED	121	Processamento concluído (inicie **dsmEndQuery**). Não há mais dados para serem processados.
DSM_RC_UNKNOWN_FORMAT	122	O arquivo que o IBM Spectrum Protect tentou restaurar ou recuperar tem um formato desconhecido.
DSM_RC_COMM_PROTOCOL_ERROR	136	Erro do protocolo de comunicação.
DSM_RC_NULL_DATABLKPTR	2001	O ponteiro não está apontando para um bloco de dados.
DSM_RC_INVALID_MCNAME	2025	O nome da classe de gerenciamento é inválido.
DSM_RC_BAD_CALL_SEQUENCE	2041	A seqüência de chamadas é inválida.
DSM_RC_WRONG_VERSION_PARM	2065	A versão do API do aplicativo cliente é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_MORE_DATA	2200	Há mais dados a serem obtidos.
DSM_RC_BUFF_TOO_SMALL	2210	O buffer é muito pequeno.

dsmGetObj

A chamada de função **dsmGetObj** obtém os dados do objeto solicitado a partir do fluxo de dados do IBM Spectrum Protect e os coloca no buffer do responsável pela chamada. A chamada de **dsmGetObj** utiliza o ID do objeto para obter o próximo objeto ou o objeto parcial do fluxo de dados.

Os dados para o objeto indicado são colocados no buffer para o qual **DataBlk** aponta. Se dados adicionais estiverem disponíveis, você deve fazer uma ou mais chamadas para **dsmGetData** para receber os dados de objetos remanescentes até um código de retorno DSM_RC_FINISHED ser retornado. Verifique o campo numBytes em **DataBlk** para ver se algum dado permanece no buffer.

Os objetos devem ser solicitados na ordem de listagem na chamada de **dsmBeginGetData** no parâmetro **dsmGetList**. A exceção é quando o cliente aplicativo precisa transmitir um objeto do fluxo de dados para chegar a um objeto posterior na lista. Se o objeto indicado pelo ID do objeto não for o próximo objeto o fluxo, o fluxo de dados é processado até o objeto ser localizado ou o fluxo ser concluído. Utilize esse recurso com cuidado, pois pode ser necessário processar e descartar grandes quantidades de dados para localizar o objeto solicitado.

Exigência: Se **dsmGetObj** retornar um código de falha (NOT FINISHED ou MORE_DATA), a sessão deverá ser encerrada para parar a operação de restauração. Isso é especialmente importante ao usar criptografia e receber RC_ENC_WRONG_KEY. Deve-se iniciar uma nova sessão com a chave adequada.

Sintaxe

```
dsInt16_t dsmGetObj  (dsUint32_t       dsmHandle,
   ObjID                *objIdP,
   DataBlk *dataBlkPtr);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

ObjID *objIdP (I)
Um ponteiro para o ID do objeto a restaurar.

DataBlk *dataBlkPtr (I/O)
Um ponteiro para o buffer onde os dados restaurados são colocados.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 36. Códigos de Retorno para dsmGetObj

Código de retorno	Explicação
DSM_RC_ABORT_INVALID_OFFSET (33)	O deslocamento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto.
DSM_RC_ABORT_INVALID_LENGTH (34)	O comprimento especificado durante uma recuperação parcial do objeto é maior do que o comprimento do objeto ou o deslocamento mais o comprimento se estende além do final do objeto.
DSM_RC_FINISHED (121)	Processamento concluído (inicie **dsmEndGetObj**).
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_MORE_DATA (2200)	Há mais dados a serem obtidos.
RC_ENC_WRONG_KEY (4580)	A chave fornecida na chamada de **dsmInitEx**, ou a chave salva, não corresponde à chave que foi utilizada para criptografar o objeto. Encerrar a sessão e fornecer a chave apropriada.

dsmGroupHandler

A chamada de função **dsmGroupHandler** executa uma ação em um grupo de arquivos lógicos, dependendo da entrada fornecida. O cliente relaciona vários objetos individuais juntos para fazer referenciar e gerenciar no servidor IBM Spectrum Protect como um grupo lógico.

Para obter informações adicionais, consulte “Agrupamento de Arquivos” na página 65.

Sintaxe

```
dsInt16_t dsmGroupHandler (dsmGroupHandlerIn_t   *dsmGroupHandlerInP,
                           dsmGroupHandlerOut_t  *dsmGroupHandlerOutP);
```

Parâmetros

dsmGroupHandlerIn_t *dsmGroupHandlerInP (I)
Transmite atributos de grupos à API.

groupType
O tipo do grupo. Os valores incluem:
- DSM_GROUPTYPE_PEER - grupo do mesmo nível

actionType
A ação a ser executada. Os valores incluem:

- DSM_GROUP_ACTION_OPEN - cria um novo grupo
- DSM_GROUP_ACTION_CLOSE - confirma e salva um grupo aberto
- DSM_GROUP_ACTION_ADD - conecta-se a um grupo
- DSM_GROUP_ACTION_ASSIGNTO - designa a outro grupo
- DSM_GROUP_ACTION_REMOVE- remove um membro de um grupo

memberType.
O tipo de grupo do objeto. Os valores incluem:
- DSM_MEMBERTYPE_LEADER - líder do grupo
- DSM_MEMBERTYPE_MEMBER - membro do grupo

***uniqueGroupTagP**
Um ID de cadeia exclusivo associado a um grupo.

leaderObjId
O ID do Objeto para o líder do grupo.

***objNameP**
Um ponteiro para o nome do objeto do líder do grupo.

memberObjList
Uma lista de objetos a remover ou designar.

dsmGroupHandlerOut_t *dsmGroupHandlerOutP (O)
Transmite o endereço da estrutura que a API conclui. O número da versão da estrutura é retornado.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 37. Códigos de Retorno para dsmGroupHandler

Código de retorno	Explicação
DSM_RC_ABORT_INVALID_GROUP_ACTION (237)	Ocorreu uma tentativa de operação inválida em um líder ou membro de grupo.

dsmInit

A chamada de função **dsmInit** inicia uma sessão de API e conecta o cliente ao armazenamento IBM Spectrum Protect. O cliente aplicativo pode ter somente uma sessão ativa por vez. Para abrir outra sessão com diferentes parâmetros, utilize primeiro a chamada de **dsmTerminate** para encerrar a sessão atual.

Para permitir consulta entre nós e restauração ou recuperação, utilize as opções de cadeia *-fromnode* e *-fromowner* . Consulte "Acessando Objetos Através de Nós e Proprietários" na página 25 para obter mais informações.

Sintaxe

```
dsInt16_t  dsmInit  (dsUint32_t        *dsmHandle,
   dsmApiVersion         *dsmApiVersionP,
   char                  *clientNodeNameP,
   char                  *clientOwnerNameP,
   char                  *clientPasswordP,
   char                  *applicationType,
   char                  *configfile,
   char          *options);
```

Parâmetros

dsUint32_t *dsmHandle (O)
O identificador que identifica essa sessão de inicialização e a associa a chamadas subsequentes da IBM Spectrum Protect.

dsmApiVersion *dsmApiVersionP (I)
Um ponteiro para a estrutura de dados que identifica a versão da API que o cliente aplicativo está utilizando para a sessão. A estrutura contém os valores das três constantes, DSM_API_VERSION, DSM_API_RELEASE e DSM_API_LEVEL, que são enviadas no arquivo dsmapitd.h . Uma chamada anterior para **dsmQueryApiVersion** deve ser executada para assegurar que existe compatibilidade entre a versão da API do cliente aplicativo e a versão da biblioteca da API instalada na estação de trabalho do usuário.

char *clientNodeNameP (I)
Esse parâmetro é um ponteiro para o nó para a sessão do IBM Spectrum Protect. Todas as sessões devem ter um nome de nó associado a ele. A constante DSM_MAX_NODE_LENGTH no arquivo dsmapitd.h configura o tamanho máximo permitido para um nome de nó.

O nome do nó não faz distinção entre maiúsculas e minúsculas.

Se esse parâmetro for configurado para NULL e *passwordaccess* for configurado para *prompt*, a API tentará obter primeiro o nome do nó da cadeia de opções transmitida. Se não estiver lá, a API tenta obter o nome do nó do arquivo de configuração ou dos arquivos de opções. Se essas tentativas para localizar o nome do nó falharem, a API do UNIX ou Linux utilizará o nome do host do sistema, enquanto que as APIs de outros sistemas operacionais retornarão o código DSM_RC_REJECT_ID_UNKNOWN code.

Esse parâmetro deve ser NULL se a opção *passwordaccess* no arquivo dsm.sys file estiver configurada para *generate*. A API utiliza o nome do host do sistema.

char *clientOwnerNameP (I)
Esse parâmetro é um ponteiro para o proprietário da sessão do IBM Spectrum Protect. Se o sistema operacional no qual a sessão é iniciada for um sistema operacional multiusuário, um nome de proprietário igual a NULL (o usuário root) tem autoridade para fazer backup, arquivar, restaurar ou recuperar qualquer objeto pertencente ao aplicativo, independentemente do proprietário do objeto.

O nome do proprietário faz distinção entre maiúsculas e minúsculas.

Esse parâmetro deve ser NULL se a opção *passwordaccess* no arquivo dsm.sys file estiver configurada para *generate*. A API utiliza então, o ID de usuário para login.

Nota: Em um sistema operacional multiusuário, se *passwordaccess* estiver configurado para *prompt*, não é necessário que o nome do proprietário corresponda ao ID de usuário ativo da sessão que está executando o aplicativo.

char *clientPasswordP (I)
Esse parâmetro é um ponteiro para a senha do nó no qual a sessão do IBM Spectrum Protect é executada. A constante DSM_MAX_VERIFIER_LENGTH no arquivo dsmapitd.h configura o tamanho máximo permitido para uma senha.

A senha não é sensível a maiúscula/minúscula.

Exceto quando o arquivo de senha é iniciado pela primeira vez, o valor desse parâmetro é ignorado se *passwordaccess* for configurado para *generate*.

char *applicationType (I)
Esse parâmetro identifica o aplicativo que está executando a sessão. O cliente aplicativo define o valor.

Toda vez que um cliente aplicativo da API inicia uma sessão com o servidor, o tipo de aplicativo (ou plataforma) do cliente é atualizado no servidor. Recomendamos que o valor do tipo do aplicativo contenha uma abreviação do sistema operacional, pois esse valor é digitado no campo de **plataforma** do servidor. O comprimento máximo da cadeia é DSM_MAX_PLATFORM_LENGTH.

Para ver o valor atual do tipo do aplicativo, chame **dsmQuerySessInfo**.

char *configfile (I)
Esse parâmetro aponta para uma cadeia de caracteres que contém o nome completo de um arquivo de configuração da API. As opções especificadas no arquivo de configuração da API substituem sua especificação no arquivo de opções do cliente. Os arquivos de opções são definidos quando o IBM Spectrum Protect (cliente ou API) é instalado.

char *options (I)
Aponta para uma cadeia de caracteres que pode conter opções do usuário, como:

- *Compressalways*
- *Servername* (somente UNIX ou Linux)
- *TCPServeraddr*
- *Fromnode*
- *Fromowner*
- *EnableClientEncryptKey*

O cliente aplicativo pode utilizar a lista de opções para substituir os valores dessas opções que o arquivo de configuração define.

O formato das opções é:
1. Cada opção especificada na lista de opções começa com um traço (-) e é seguida pela palavra-chave da opção.
2. A palavra-chave, por sua vez, é seguida por um sinal de igual (=) e pelo parâmetro da opção.
3. Se o parâmetro da opção contiver um espaço em branco, coloque o parâmetro entre aspas simples ou duplas.
4. Se mais de uma opção for especificada, separe as opções por espaços em branco.

Se as opções forem NULL, os valores para todas as opções são retirados do arquivo de opções do usuário ou do arquivo de configuração da API.

Códigos de Retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 38. Códigos de Retorno para dsmInit

Código de retorno	Explicação
DSM_RC_ABORT_SYSTEM_ERROR (1)	O servidor detectou um erro no sistema e notificou os clientes.
DSM_RC_REJECT_VERIFIER_EXPIRED (52)	A senha expirou e deve ser atualizada.
DSM_RC_REJECT_ID_UNKNOWN (53)	Não foi possível localizar o nome do nó.

Tabela 38. Códigos de Retorno para dsmInit (continuação)

Código de retorno	Explicação
DSM_RC_AUTH_FAILURE (137)	Ocorreu uma falha de autenticação.
DSM_RC_NO_STARTING_DELIMITER (148)	Não há nenhum delimitador inicial no padrão.
DSM_RC_NEEDED_DIR_DELIMITER (149)	Um delimitador de diretório é necessário imediatamente antes e após a meta cadeia "diretórios correspondentes" ("...") e uma que não foi localizada.
DSM_RC_NO_PASS_FILE (168)	O arquivo de senha não está disponível.
DSM_RC_UNMATCHED_QUOTE (177)	Aspas sem correspondente se encontra na cadeia de opções.
DSM_RC_NLS_CANT_OPEN_TXT (0610)	Não foi possível abrir o arquivo de texto de mensagens.
DSM_RC_INVALID_OPT (400)	Uma entrada na cadeia de opções é inválida.
DSM_RC_INVALID_DS_HANDLE (2014)	Identificador DSM inválido.
DSM_RC_NO_OWNER_REQD (2032)	O parâmetro do proprietário deve ser NULL quando *passwordaccess* estiver configurado para *generate*.
DSM_RC_NO_NODE_REQD (2033)	O parâmetro do nó deve ser NULL quando *passwordaccess* estiver configurado para *generate*.
DSM_RC_WRONG_VERSION (2064)	A versão da API para o cliente aplicativo tem um valor mais alto do que a versão do IBM Spectrum Protect.
DSM_RC_PASSWD_TOOLONG (2103)	A senha especificada é muito longa.
DSM_RC_NO_OPT_FILE (2220)	Um arquivo de configuração não pôde ser localizado.
DSM_RC_INVALID_KEYWORD (2221)	Uma palavra-chave especificada em uma cadeia de opções é inválida.
DSM_RC_PATTERN_TOO_COMPLEX (2222)	O padrão incluir-excluir é muito complexo para ser interpretado pelo IBM Spectrum Protect.
DSM_RC_NO_CLOSING_BRACKET (2223)	Está faltando o colchete direito no padrão.
DSM_RC_INVALID_SERVER (2225)	Para um ambiente multiusuário, o servidor não foi localizado no arquivo de configuração do sistema.
DSM_RC_NO_HOST_ADDR (2226)	Não há informações suficientes para conectar ao host.
DSM_RC_MACHINE_SAME (2227)	O nome do nó definido no arquivo de opções não pode ser o mesmo que o nome do host do sistema.
DSM_RC_NO_API_CONFIGFILE (2228)	Não é possível abrir o arquivo de configuração.
DSM_RC_NO_INCLEXCL_FILE (2229)	O arquivo include-exclude não foi localizado.
DSM_RC_NO_SYS_OR_INCLEXCL (2230)	O arquivo dsm.sys ou o arquivo include-exclude não foi localizado.

Conceitos relacionados:

- Visão geral do arquivo de opções do cliente
- Opções de processamento

dsmInitEx

A chamada de função **dsmInitEx** inicia uma sessão de API usando os parâmetros adicionais para verificação estendida.

Sintaxe

```
dsInt16_t  dsmInitEx   (dsUint32_t        *dsmHandleP,
                        dsmInitExIn_t         *dsmInitExInP,
                        dsmInitExOut_t    *dsmInitExOutP) ;
```

Parâmetros

dsUint32_t *dsmHandleP (O)
O identificador que identifica essa sessão de inicialização e a associa a chamadas subsequentes da IBM Spectrum Protect.

dsmInitExIn_t *dsmInitExInP
Essa estrutura contém os seguintes parâmetros de entrada:

dsmApiVersion *dsmApiVersionP (I)
Esse parâmetro é um ponteiro para a estrutura de dados que identifica a versão da API que o cliente aplicativo está utilizando para a sessão. A estrutura contém os valores das quatro constantes, DSM_API_VERSION, DSM_API_RELEASE, DSM_API_LEVEL e DSM_API_SUBLEVEL, que estão configuradas no arquivo dsmapitd.h. Chame **dsmQueryApiVersionEx** e verifique se a versão da API do aplicativo cliente e a versão da biblioteca da API instalada na estação de trabalho do usuário são compatíveis.

char *clientNodeNameP (I)
Esse parâmetro é um ponteiro para o nó para a sessão do IBM Spectrum Protect. Todas as sessões devem estar associadas a um nome do nó. A constante DSM_MAX_NODE_LENGTH no arquivo dsmapitd.h configura o tamanho máximo para um nome do nó.

O nome do nó não faz distinção entre maiúsculas e minúsculas.

Se esse parâmetro for configurado para NULL e **passwordaccess** for configurado para prompt, a API tentará obter primeiro o nome do nó da sequência de opções transmitida. Se não estiver lá, a API tenta obter o nome do nó do arquivo de configuração ou dos arquivos de opções. Se essas tentativas de localizar o nome do nó falharem, a API do UNIX ou do Linux usará o nome do host do sistema, enquanto as APIs de outros sistemas operacionais retornam DSM_RC_REJECT_ID_UNKNOWN.

Esse parâmetro deve ser NULL se a opção **passwordaccess** no arquivo dsm.sys estiver configurada para generate. A API então utiliza o nome do host do sistema.

char *clientOwnerNameP (I)
Esse parâmetro é um ponteiro para o proprietário da sessão do IBM Spectrum Protect. Se o sistema operacional for uma plataforma multiusuário, um nome de proprietário NULL (o usuário raiz) tem autoridade para fazer backup, arquivar, restaurar ou recuperar qualquer objeto pertencente ao aplicativo, independentemente do proprietário do objeto.

O nome do proprietário faz distinção entre maiúsculas e minúsculas.

Esse parâmetro deve ser NULL se a opção **passwordaccess** no arquivo dsm.sys estiver configurada para generate. A API utiliza então, o ID de usuário para login.

Dica: Em uma plataforma multiusuário, se **passwordaccess** estiver configurado para `prompt`, não será necessário que o nome do proprietário corresponda ao ID de usuário ativo da sessão que está executando o aplicativo.

char *clientPasswordP (I)
Um ponteiro para a senha do nó no qual a sessão do IBM Spectrum Protect é executada. A constante DSM_MAX_VERIFIER_LENGTH no arquivo `dsmapitd.h` configura o tamanho máximo permitido para uma senha.

A senha não é sensível a maiúscula/minúscula.

Exceto quando o arquivo de senha é iniciado pela primeira vez, o valor desse parâmetro é ignorado se **passwordaccess** for configurado para `generate`.

char *userNameP;
Um ponteiro para o nome do usuário administrativo que tem autoridade de cliente para o nó.

char *userPasswordP;
Um ponteiro para a senha do parâmetro **userName**, se um valor for fornecido.

char *applicationType (I)
Identifica o aplicativo que está executando a sessão da IBM Spectrum Protect. O cliente aplicativo identifica o valor.

Toda vez que um cliente aplicativo da API inicia uma sessão com o servidor, o tipo de aplicativo (ou sistema operacional) do cliente é atualizado no servidor. O valor é inserido no campo de **plataform** do servidor. Considere usar um ID do sistema operacional no valor. O comprimento máximo da sequência é definido na constante DSM_MAX_PLATFORM_LENGTH.

Para visualizar o valor atual do tipo do aplicativo, chame **dsmQuerySessInfo**.

char *configfile (I)
Aponta para uma sequência de caracteres que contém o nome completo de um arquivo de configuração da API. As opções especificadas no arquivo de configuração da API substituem sua especificação no arquivo de opções do cliente. Os arquivos de opções são definidos quando o IBM Spectrum Protect (cliente ou API) é instalado.

char *options (I)
Aponta para uma cadeia de caracteres que pode conter opções do usuário, como:
- `COMPRESSALWAYS`
- `Servername` (Somente sistemas UNIX e Linux)
- `TCPServeraddr` (não se aplica a sistemas UNIX)
- `Fromnode`
- `Fromowner`

O cliente aplicativo pode utilizar a lista de opções para substituir os valores dessas opções que o arquivo de configuração define.

As opções têm o formato a seguir:
1. Cada opção especificada na lista de opções começa com um traço (-) e é seguida pela palavra-chave da opção.
2. A palavra-chave é seguida por um sinal de igual (=) e pelo parâmetro da opção.

3. Se o parâmetro da opção contiver um espaço em branco, coloque o parâmetro entre aspas simples ou duplas.
4. Se mais de uma opção for especificada, separe as opções por espaços em branco.

Se as opções forem NULL, os valores para todas as opções serão retirados do arquivo de opções do usuário ou do arquivo de configuração da API.

dirDelimiter
O delimitador de diretório utilizado como prefixo nos nomes de nível superior ou nível inferior do espaço no arquivo. Você deve especificar o parâmetro **dirDelimiter** somente se o aplicativo substituir os padrões do sistema. Em um ambiente UNIX ou Linux, o padrão é uma barra (/). Em um ambiente Windows, o padrão é uma barra invertida (\).

useUnicode
Uma sinalização booleana que indica se o Unicode foi ativado. A sinalização **useUnicode** deve ser `false` para atingir a interoperabilidade multiplataforma entre os sistemas UNIX e Windows.

bCrossPlatform
Uma sinalização booleana que deve ser configurada (`bTrue`) para atingir a interoperabilidade multiplataforma entre os sistemas UNIX e Windows. Quando a sinalização `bCrossPlatform` for configurada, a API assegurará que os espaços no arquivo não são Unicode e que o aplicativo não usa Unicode. Um aplicativo do Windows que usa Unicode não é compatível com aplicativos que usam codificações não Unicode. A sinalização `bCrossPlatform` não deve ser configurada para um aplicativo do Windows que usa Unicode.

UseTsmBuffers
Indica se a eliminação de cópia de buffer deve ser utilizada.

numTsmBuffers
Número de buffers quando `useTsmBuffers=bTrue`.

bEncryptKeyEnabled
Indica se criptografia com chave gerenciada pelo aplicativo é utilizada.

encryptionPasswordP
A senha de criptografia.

Restrição: Quando `encryptkey=save`, se existir uma chave de criptografia, o valor especificado no **encryptionPasswordP** será ignorado.

dsmAppVersion *appVersionP (I)
Esse parâmetro é um ponteiro para a estrutura de dados que identifica as informações de versão do aplicativo que está iniciando uma sessão da API. A estrutura contém os valores das quatro constantes, applicationVersion, applicationRelease, applicationLevel e applicationSubLevel, que estão configuradas no arquivo `tsmapitd.h`.

dsmInitExOut_t *dsmInitExOut P
Essa estrutura contém os parâmetros de saída.

dsUint32_t *dsmHandle (0)
O identificador que identifica esta sessão de inicialização e a associa a chamadas subsequentes da API.

infoRC
Informações adicionais sobre o código de retorno. Verifique o código de retorno de função e o valor de **infoRC**. Um valor **infoRC** de

DSM_RC_REJECT_LASTSESS_CANCELED (69), o IBM Spectrum Protect indica que o administrador cancelou a última sessão.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 39. Códigos de Retorno para **dsmInitEx**

Código de retorno	Explicação
DSM_RC_ABORT_SYSTEM_ERROR (1)	O servidor IBM Spectrum Protect detectou um erro no sistema e notificou os clientes.
DSM_RC_REJECT_VERIFIER_EXPIRED (52)	A senha expirou e deve ser atualizada. A próxima chamada deve ser **dsmChangePW** com o identificador retornado na chamada.
DSM_RC_REJECT_ID_UNKNOWN (53)	Não é possível localizar o nome do nó.
DSM_RC_TA_COMM_DOWN (103)	O link de comunicações está inativo.
DSM_RC_AUTH_FAILURE (137)	Ocorreu uma falha de autenticação.
DSM_RC_NO_STARTING_DELIMITER (148)	Não há nenhum delimitador inicial no padrão.
DSM_RC_NEEDED_DIR_DELIMITER (149)	Um delimitador de diretório é necessário imediatamente antes e depois da meta cadeia dos "diretórios correspondentes" ("..."), mas não foi localizado.
DSM_RC_NO_PASS_FILE (168)	O arquivo de senha não está disponível.
DSM_RC_UNMATCHED_QUOTE (177)	Aspas sem correspondente estão na sequência de opções.
DSM_RC_NLS_CANT_OPEN_TXT (0610)	Não foi possível abrir o arquivo de texto de mensagens.
DSM_RC_INVALID_OPT (2013)	Uma entrada na cadeia de opções é inválida.
DSM_RC_INVALID_DS_HANDLE (2014)	Identificador DSM inválido.
DSM_RC_NO_OWNER_REQD (2032)	O parâmetro do proprietário deve ser NULL quando **passwordaccess** estiver configurado para `generate`.
DSM_RC_NO_NODE_REQD (2033)	O parâmetro do nó deve ser NULL quando **passwordaccess** estiver configurado para generate.
DSM_RC_WRONG_VERSION (2064)	A versão da API do cliente aplicativo tem um valor mais alto do que a versão do IBM Spectrum Protect.
DSM_RC_PASSWD_TOOLONG (2103)	A senha especificada é muito longa.
DSM_RC_NO_OPT_FILE (2220)	Nenhum arquivo de configuração foi localizado.
DSM_RC_INVALID_KEYWORD (2221)	Uma palavra-chave especificada em uma sequência de opções é inválida.
DSM_RC_PATTERN_TOO_COMPLEX (2222)	O padrão incluir-excluir é muito complexo para ser interpretado pelo IBM Spectrum Protect.
DSM_RC_NO_CLOSING_BRACKET (2223)	Está faltando o colchete direito no padrão.
DSM_RC_INVALID_SERVER (2225)	Para um ambiente multiusuário, o servidor não foi localizado no arquivo de configuração do sistema.
DSM_RC_NO_HOST_ADDR (2226)	Não há informações suficientes para conectar ao host.
DSM_RC_MACHINE_SAME (2227)	O nome do nó definido no arquivo de opções não pode ser o mesmo que o nome do host do sistema.
DSM_RC_NO_API_CONFIGFILE (2228)	Não é possível abrir o arquivo de configuração.
DSM_RC_NO_INCLEXCL_FILE (2229)	O arquivo include-exclude não foi localizado.
DSM_RC_NO_SYS_OR_INCLEXCL (2230)	`dsm.sys` ou o arquivo include-exclude não foi localizado.

Conceitos relacionados:

Visão geral do arquivo de opções do cliente

Opções de processamento

dsmLogEvent

A chamada de função **dsmLogEvent** registra uma mensagem do usuário (ANE4991 I) no arquivo de log do servidor, no log de erros local ou em ambos. Uma estrutura do tipo **logInfo** é transmitida na chamada. Essa chamada deve ser executada enquanto no estado **InSession** em uma sessão. Não a execute em send, get ou query. Para recuperar mensagens registradas no servidor, utilize o comando **query actlog** através do cliente administrativo.

Consulte o diagrama de estado resumido, Figura 20 na página 78.

Sintaxe

```
dsInt16_t dsmLogEvent
     (dsUint32_t          dsmHandle,
     logInfo        *logInfoP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

logInfo *logInfoP (I)
Transmite a mensagem e o destino. O cliente aplicativo é responsável por alocar armazenamento para a estrutura.

Os campos na estrutura **logInfo** são:

message
O texto da mensagem a ser registrada. Deve ser uma cadeia terminada em null. O comprimento máximo é DSM_MAX_RC_MSG_LENGTH.

dsmLogtype
Especifica onde registrar a mensagem. Valores possíveis incluem: **logServer**, **logLocal**, **logBoth**.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 40. Códigos de Retorno para dsmLogEvent

Código de retorno	Explicação
DSM_RC_STRING_TOO_LONG (2120)	A cadeia da mensagem é muito longa.

dsmLogEventEx

A chamada de função **dsmLogEventEx** registra uma mensagem do usuário no arquivo de log do servidor, no log de erro local ou em ambos. Esta chamada deve ser feita enquanto em um estado **InSession** em uma sessão. A chamada não pode ser feita em uma chamada de envio, obtenção ou consulta.

Diagrama de estado de resumo: Para obter uma visão geral das interações da sessão, consulte o diagrama de estado de resumo no tópico a seguir:

Figura 20 na página 78

A gravidade determina o número da mensagem do IBM Spectrum Protect. Para visualizar mensagens que estão conectadas no servidor, use o comando **query actlog** através do cliente administrativo. Use a opção do cliente IBM Spectrum Protect, `errorlogretention`, para remover o arquivo de log de erros do cliente se o aplicativo gerar várias mensagens de cliente gravadas no log do cliente `dsmLogType`, `logLocal` ou `logBoth`. Para obter mais informações, consulte a documentação do servidor IBM Spectrum Protect.

Sintaxe

```
extern dsInt16_t DSMLINKAGE  dsmLogEventEx(
       dsUint32_t             dsmHandle,
       dsmLogExIn_t          *dsmLogExInP,
       dsmLogExOut_t         *dsmLogExOutP
);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

dsmLogExIn_t *dsmLogExInP
Essa estrutura contém os parâmetros de entrada.

dsmLogSeverity severity;
Esse parâmetro é a gravidade do evento. Os valores possíveis são:

```
logSevInfo,       /* informativa ANE4990 */
logSevWarning,    /* aviso       ANE4991 */
logSevError,      /* Erro        ANE4992 */
logSevSevere      /* grave       ANE4993 */
```

char appMsgID[8];
Esse parâmetro é uma cadeia para identificar a mensagem específica do aplicativo. Um formato adequado é três caracteres que são seguidos por quatro números, por exemplo: `DSM0250`.

dsmLogType logType;
Esse parâmetro especifica para onde direcionar o evento. O parâmetro possui os valores possíveis a seguir:

- `logServer`
- `logLocal`
- `logBoth`

char *message;
Esse parâmetro é o texto da mensagem do evento a registrar. O texto deve ser uma sequência terminada em null. O comprimento máximo é `DSM_MAX_RC_MSG_LENGTH`.

Restrição: Mensagens que vão para o servidor devem estar em inglês. Mensagens que não estão no idioma inglês não são exibidas corretamente.

dsmLogExOut_t *dsmLogExOutP
Essa estrutura contém os parâmetros de saída. Atualmente, não há parâmetros de saída.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 41. Códigos de Retorno para dsmLogEventEx

Código de retorno	Explicação
DSM_RC_STRING_TOO_LONG (2120)	A cadeia da mensagem é muito longa.

dsmQueryAccess

A chamada de função **dsmQueryAccess** consulta no servidor todas as regras de autorização de acesso para versões de backup ou cópias arquivadas de seus objetos. Um ponteiro para uma matriz de regras de acesso é transmitida para a chamada e a matriz preenchida é retornada. Um ponteiro para o número de regras é transmitido para indicar quantas regras estão na matriz.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t DSMLINKAGE dsmQueryAccess
                  (dsUint32_t           dsmHandle),
                   qryRespAccessData    **accessListP,
                   dsUint16_t           *numberOfRules) ;
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

qryRespAccessData **accessListP (O)
Um ponteiro para uma matriz de elementos qryRespAccessData que a biblioteca da API aloca. Cada elemento corresponde a uma regra de acesso. O número de elementos da matriz é retornado no parâmetro **numberOfRules**. As informações retornadas em cada elemento qryRespAccessData inclui o seguinte:

Nome	Descrição
ruleNumber	O ID da regra de acesso. Isso identifica a regra para exclusão.
AccessType	O tipo de backup ou de archive.
Node	O nó no qual você forneceu acesso.
Owner	O usuário para o qual concedeu acesso.
objName	Os descritores do espaço no arquivo de nível superior ou de nível inferior.

dsUint32_t *numberOfRules (O)
Retorna o número de regras na matriz accessList.

dsmQueryApiVersion

A chamada de função **dsmQueryApiVersion** executa um pedido de consulta para a versão da biblioteca da API que o cliente aplicativo acessa.

Todas as atualizações da API são feitas em um formato com compatibilidade superior. Qualquer cliente aplicativo com uma versão ou release de API inferior ou igual ao da biblioteca da API na estação de trabalho do usuário final opera sem alteração. Esteja ciente antes de prosseguir que se a chamada de **dsmQueryApiVersion** retornar uma versão ou release de versão anterior ao dos clientes aplicativos, algumas chamadas da API podem ser aprimoradas de forma não suportada pela versão mais antiga da API do usuário final.

O número da versão da API do aplicativo é armazenado no arquivo de cabeçalho dsmapitd.h como constantes DSM_API_VERSION, DSM_API_RELEASE e DSM_API_LEVEL.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
void dsmQueryApiVersion  (dsmApiVersion *apiVersionP);
```

Parâmetros

dsmApiVersion *apiVersionP (0)
Esse parâmetro é um ponteiro para a estrutura que contém os componentes de versão, release e nível da biblioteca da API. Por exemplo, se a versão da biblioteca for 1.1.0, então, após retornar da chamada, os campos da estrutura conterão os seguintes valores:

```
dsmApiVersionP->version  = 1
dsmApiVersionP->release  = 1
dsmApiVersionP->level    = 0
```

dsmQueryApiVersionEx

A chamada de função **dsmQueryApiVersionEx** executa um pedido de consulta para a versão da biblioteca da API que o cliente aplicativo acessa.

Todas as atualizações da API são feitas em um formato com compatibilidade superior. Qualquer cliente aplicativo que tem uma versão ou release de API inferior ou igual ao da biblioteca da API na estação de trabalho do usuário final opera sem alteração. Consulte Resumo de Alterações de Código no arquivo README_api_enu para obter exceções da compatibilidade superior. Se a chamada de **dsmQueryApiVersionEx** retornar uma versão ou release de versão diferentes do cliente aplicativo, esteja ciente antes de prosseguir que algumas chamadas da API podem ser aprimoradas de forma não suportada pela versão mais antiga da API do usuário final.

O número da versão da API do aplicativo é armazenado no arquivo de cabeçalho dsmapitd.h como constantes DSM_API_VERSION, DSM_API_RELEASE, DSM_API_LEVEL e DSM_API_SUBLEVEL.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
void dsmQueryApiVersionEx (dsmApiVersionEx *apiVersionP);
```

Parâmetros

dsmApiVersionEx *apiVersionP (O)

Esse parâmetro é um ponteiro para a estrutura que contém os componentes de versão, release, nível e subnível da biblioteca da API. Por exemplo, se a biblioteca for da Versão 5.5.0.0, após retornarem da chamada, os campos da estrutura conterão os seguintes valores:

- `ApiVersionP->version  = 5`
- `ApiVersionP->release  = 5`
- `ApiVersionP->level    = 0`
- `ApiVersionP->subLevel = 0`

dsmQueryCliOptions

A chamada de função **dsmQueryCliOptions** consulta valores de opções importantes nos arquivos de opções do usuário. Uma estrutura do tipo **optStruct** é transmitida na chamada e contém as informações. Essa chamada é executada antes de **dsmInitEx** ser chamada e determina a configuração antes da sessão.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t dsmQueryCliOptions
    (optStruct     *optstructP);
```

Parâmetros

optStruct *optstructP (I/O)

Esse parâmetro transmite o endereço da estrutura que a API conclui. O cliente aplicativo é responsável por alocar armazenamento para a estrutura. Em um retorno bem-sucedido, as informações apropriadas são digitadas nos campos na estrutura.

As informações a seguir são retornadas na estrutura **optStruct**:

Nome	Descrição
dsmiDir	O valor da variável de ambiente DSMI_DIR.
dsmiConfig	O arquivo de opções do cliente, conforme especificação pela variável de ambiente DSMI_CONFIG.
serverName	O nome do servidor IBM Spectrum Protect.
commMethod	O método de comunicação selecionado. Consulte #defines para DSM_COMM_* no arquivo dsmapitd.h.
serverAddress	O endereço do servidor baseado no método de comunicação.
nodeName	O nome do nó cliente (máquina).
compression	Este campo fornece informações sobre a opção de compactação.
passwordAccess	Os valores são: *bTrue* para generate e *bFalse* para prompt.

Conceitos relacionados:

Opções de processamento

dsmQuerySessInfo

A chamada de função **dsmQuerySessInfo** inicia uma solicitação de consulta para o IBM Spectrum Protect para obter informações relacionadas à operação da sessão especificada em **dsmHandle**. Uma estrutura do tipo **ApiSessInfo** é transmitida na chamada com todas as informações relacionadas à sessão disponíveis digitadas. Essa chamada é iniciada após uma chamada de **dsmInitEx** bem-sucedida.

As informações retornadas na estrutura **ApiSessInfo** incluem o seguinte:
- Informações do servidor: número da porta, data e hora e tipo
- Padrões do cliente: tipo de aplicativo, permissões de exclusão, delimitadores e limites da transação
- Informações de sessão: ID de login e proprietário
- Dados da política: domínio, conjunto de políticas ativo e período de tolerância de retenção

Consulte o Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163 para obter informações sobre o conteúdo da estrutura que é transmitida e cada campo dela.

Sintaxe

```
dsInt16_t dsmQuerySessInfo  (dsUint32_t       dsmHandle,
   ApiSessInfo   *SessInfoP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

ApiSessInfo *SessInfoP (I/O)
Esse parâmetro transmite o endereço da estrutura que a API insere. O cliente aplicativo é responsável por alocar armazenamento para a estrutura e por preencher as entradas de campo que indicam a versão da estrutura utilizada. Em um retorno bem-sucedido, os campos da estrutura são preenchidos com as informações apropriadas. O **adsmServerName** é o nome fornecido no comando **define server** no servidor IBM Spectrum Protect. Se o campo archiveRetentionProtection for verdadeiro, o servidor é ativado para proteção da retenção.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 42. Códigos de Retorno para dsmQuerySessInfo

Código de retorno	Explicação
DSM_RC_NO_SESS_BLK (2006)	Nenhuma informação do bloco de sessão do servidor.
DSM_RC_NO_POLICY_BLK (2007)	Nenhuma informação da política do servidor disponível.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.

dsmQuerySessOptions

A chamada de função **dsmQuerySessOptions** consulta valores de opções importantes que são válidos na sessão especificada em **dsmHandle**. Uma estrutura do tipo optStruct é transmitida na chamada e contém as informações.

Essa chamada é iniciada após uma chamada de **dsmInitEx** bem-sucedida. Os valores retornados podem ser diferentes dos valores retornados em uma chamada de **dsmQueryCliOptions**, dependendo dos valores que são transmitidos para a chamada de **dsmInitEx**, principalmente optString, e optFile. Para obter informações sobre a precedência das opções, consulte "Entendendo os Arquivos de Configuração e Opções" na página 1.

Não há códigos de retorno específicos para essa chamada.

Sintaxe

```
dsInt16_t dsmQuerySessOptions
    (dsUint32_t          dsmHandle,
    optStruct      *optstructP);
```

Parâmetros

dsUint32_t dsmhandle(I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

optStruct *optstructP (I/O)
Esse parâmetro transmite o endereço da estrutura que a API conclui. O cliente aplicativo é responsável por alocar armazenamento para a estrutura. Em um retorno bem-sucedido, os campos da estrutura são preenchidos com as informações apropriadas.

As informações retornadas na estrutura optStruct são:

Nome	Descrição
dsmiDir	O valor da variável de ambiente DSMI_DIR.
dsmiConfig	O arquivo dsm.opt que a variável de ambiente DSMI_CONFIG especifica.
serverName	O nome da sub-rotina do servidor IBM Spectrum Protect no arquivo de opções.
commMethod	O método de comunicação selecionado. Consulte #defines para DSM_COMM_* no arquivo dsmapitd.h.
serverAddress	O endereço do servidor baseado no método de comunicação.
nodeName	O nome do nó cliente (máquina).
compressão	O valor da opção de compactação (bTrue=on e bFalse=off).
compressAlways	O valor da opção compressalways (bTrue=on e bFalse=off).
passwordAccess	O valor bTrue para generate e bFalse para prompt.

Conceitos relacionados:

Opções de processamento

dsmRCMsg

A chamada de função **dsmRCMsg** obtém o texto de mensagem que está associado a um código de retorno da API.

O parâmetro **msg** exibe o código de retorno do prefixo da mensagem entre parênteses ( ), seguido pelo texto da mensagem. Por exemplo, uma chamada a **dsmRCMsg** pode retornar o seguinte:

```
ANS0264E (RC2300) Somente o usuário root pode executar dsmChangePW ou
                          dsmDeleteFS.
```

Para alguns idiomas nos quais os caracteres são diferentes nas páginas de códigos ANSII e OEM, pode ser necessário converter cadeias de ANSII para OEM antes de imprimi-las (por exemplo, conjuntos de caracteres de byte único do leste europeu). Segue um exemplo:

```
dsmRCMsg(dsmHangle, rc, msgBuf);
#ifdef WIN32
#ifndef WIN64
CharToOemBuff(msgBuf, msgBuf, strlen(msgBuf));
#endif
#endif
printf("
```

Sintaxe

```
dsInt16_t dsmRCMsg  (dsUint32_t       dsmHandle,
   dsInt16_t             dsmRC,
   char        *msg);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsInt16_t dsmRC (I)
O código de retorno da API do texto da mensagem associado. Os códigos de retorno da API são listados no arquivo dsmrc.h. Consulte Apêndice A, "Arquivo de Origem dos Códigos de Retorno do API : dsmrc.h", na página 153 para obter mais informações.

char *msg (O)
Esse parâmetro é o texto da mensagem associado ao código de retorno, **dsmRC**. O responsável pela chamada é responsável por alocar espaço suficiente para o texto da mensagem.

O comprimento máximo para **msg** é definido como DSM_MAX_RC_MSG_LENGTH.

Nas plataformas que possuem Suporte ao Idioma Nacional e uma opção de arquivos de mensagens de idioma, a API retorna uma cadeia de mensagem no idioma nacional.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 43. Códigos de Retorno para dsmRCMsg

Código de retorno	Explicação
DSM_RC_NULL_MSG (2002)	O parâmetro **msg** para a chamada de dsmRCMsg é um ponteiro NULL.

Tabela 43. Códigos de Retorno para dsmRCMsg (continuação)

Código de retorno	Explicação
DSM_RC_INVALID_RETCODE (2021)	O código de retorno transmitido à chamada de **dsmRCMsg** é um código inválido.
DSM_RC_NLS_CANT_OPEN_TXT (0610)	Não foi possível abrir o arquivo de texto de mensagens.

dsmRegisterFS

A chamada de função **dsmRegisterFS** registra um novo espaço no arquivo com o servidor IBM Spectrum Protect. Primeiro, registre um espaço no arquivo antes de poder fazer backup dos dados.

Os clientes aplicativos não devem utilizar os mesmos nomes de espaço no arquivo que um cliente de backup/archive utilizaria.

- No UNIX ou Linux, execute o comando **df** para esses nomes.
- No Windows, esses nomes geralmente são os rótulos de volume que estão associados com unidades diferentes em seu sistema.

Sintaxe

```
dsInt16_t dsmRegisterFS  (dsUint32_t       dsmHandle,
   regFSData     *regFilespaceP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

regFSData *regFilespaceP (I)
O parâmetro transmite o nome do espaço no arquivo e as informações associadas que você precisa registrar com o servidor IBM Spectrum Protect.

Dica: O campo *fstype* inclui o prefixo, **"API:".** Todas as consultas do espaço no arquivo exibem essa cadeia. Por exemplo, se o usuário transmitir *myfstype* para *fstype* em**dsmRegisterFS**, a cadeia de valor real no servidor será retornada como `API:myfstype` quando consultada. Esse prefixo distingue os objetos de API dos objetos de backup/archive.

A área útil para **fsInfo** agora é DSM_MAX_USER_FSINFO_LENGTH.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 44. Códigos de Retorno para dsmRegisterFS

Código de retorno	Explicação
DSM_RC_INVALID_FSNAME (2016)	O nome do espaço no arquivo é inválido.
DSM_RC_INVALID_DRIVE_CHAR (2026)	A letra da unidade não é um caractere alfabético.
DSM_RC_NULL_FSNAME (2027)	O nome do espaço no arquivo é nulo.
DSM_RC_FS_ALREADY_REGED (2062)	O espaço no arquivo já está registrado.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_FSINFO_TOOLONG (2106)	As informações do espaço no arquivo são muito longas.

dsmReleaseBuffer

A função **dsmReleaseBuffer** retorna um buffer ao IBM Spectrum Protect. O aplicativo chama **dsmReleaseBuffer** após **dsmGetDataEx** ter sido chamada e o aplicativo ter movido todos os dados para fora do buffer e estar pronto para liberá-los. **dsmReleaseBuffer** requer que **dsmInitEx** tenha sido chamado com ***UseTsmBuffers*** configurado para *btrue* e um valor diferente de zero tenha sido fornecido para *numTsmBuffers*. **dsmReleaseBuffer** também deve ser chamado se o aplicativo está prestes a chamar **dsmTerminate** e ele ainda contém buffers de dados.

dsmReleaseBufferSyntax

```
dsInt16_t  dsmReleaseBuffer   (releaseBufferIn_t        *dsmReleaseBufferInP,
                               releaseBufferOut_t       *dsmReleaseBufferOutP) ;
```

Parâmetros

releaseBufferIn_t * dsmReleaseBufferInP (I)
Essa estrutura contém os seguintes parâmetros de entrada.

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

dsUint8_t tsmBufferHandle(I)
O identificador que identifica o buffer.

char *dataPtr(I)
O endereço no qual o aplicativo é gravado.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 45. Códigos de Retorno para dsmReleaseBuffer

Código de retorno	Explicação
DSM_RC_BAD_CALL_SEQUENCE	A chamada não foi emitida no estado apropriado.
DSM_RC_INVALID_TSMBUFFER	O identificador ou o valor de **dataPtr** é inválido.
DSM_RC_BUFF_ARRAY_ERROR	Ocorreu um erro de matriz do buffer.

dsmRenameObj

A chamada de função **dsmRenameObj** renomeia o nome do objeto de nível superior ou de nível inferior. Para objetos de backup, transmita o nome do objeto atual e alterações para os nomes de objetos de nível superior ou de nível inferior. Para objetos de archive, transmita o nome do espaço no arquivo do objeto atual e o ID do objeto e alterações para os nomes de objetos de nível superior ou de nível inferior. Utilize essa chamada de função nas chamadas de **dsmBeginTxn** e **dsmEndTxn**.

O sinalizador de associação determina se um nome de objeto de backup duplicado deve ou não ser associado com os backups existentes. Se o novo nome corresponder a um objeto existente e a associação for verdadeira, o objeto atual é convertido para o novo nome e torna-se a versão ativa do novo nome enquanto o objeto ativo existente que tinha o mesmo nome torna-se a cópia inativa mais

superior do objeto. Se o novo nome corresponder a um objeto existente e a associação for falsa, a função retorna o código de retorno, DSM_RC_ABORT_DUPLICATE_OBJECT.

Restrição: Somente o proprietário do objeto pode renomeá-lo.

A chamada de função **dsmRenameObj** testa as seguintes condições de associação:

- O objeto **dsmObjName** atual e o novo objeto de nível superior ou de nível inferior devem corresponder quanto a proprietário, grupo de cópias e classe de gerenciamento.
- O **dsmObjName** atual deve ter feito backup mais recentemente do que o objeto ativo atualmente com o novo nome.
- Deve haver somente uma cópia ativa do **dsmObjName** atual sem nenhuma cópia inativa.

Sintaxe

```
dsInt16_t dsmRenameObj  (dsmRenameIn_t       *dsmRenameInP,
                         dsmRenameOut_t      *dsmRenameOutP);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmRenameIn_t *dsmRenameInP
Essa estrutura contém os parâmetros de entrada.

dsUint8_t repository (I);
Esse parâmetro indica se o espaço no arquivo a ser excluído está no repositório de backup ou no repositório de archive.

dsmObjName *objNameP (I);
Esse parâmetro é um ponteiro para a estrutura que contém o nome do espaço no arquivo atual, o nome do objeto de nível superior, o nome do objeto de nível inferior e o tipo de objeto.

char newHl [DSM_MAX_HL_LENGTH + 1];
Esse parâmetro especifica o novo nome de nível superior.

char newLl [DSM_MAX_LL_LENGTH + 1];
Esse parâmetro especifica o novo nome de nível inferior.

dsBool_t merge;
Esse parâmetro determina se um objeto de backup está associado com objetod denominados duplicados. Os valores são true ou false.

ObjID;
O ID do objeto para objetos de archive.

dsmRenameOut_t *dsmRnameOutP
Essa estrutura contém os parâmetros de saída.

Nota: Não há parâmetros de saída.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 46. Códigos de Retorno para dsmRenameObj

Código de retorno	Explicação
DSM_RC_ABORT_MERGE_ERROR (45)	O servidor detectou um erro de associação.
DSM_RC_ABORT_DUPLICATE_OBJECT (32)	O objeto já existe e a associação é falsa.
DSM_RC_ABORT_NO_MATCH (2)	Objeto não localizado.
DSM_RC_REJECT_SERVER_DOWNLEVEL (58)	O servidor IBM Spectrum Protect deve estar na V3.7.4.0 ou posterior para que essa função funcione.

dsmRequestBuffer

A função **dsmRequestBuffer** retorna um buffer para o IBM Spectrum Protect. O aplicativo chama **dsmRequestBuffer** após uma **dsmGetDataEx** ter sido chamada e o aplicativo ter movido todos os dados para fora do buffer e estar pronto para liberá-lo.

dsmReleaseBuffer requer que **dsmInitEx** tenha sido chamado com *UseTsmBuffers* configurado para *btrue* e um valor que não seja zero tenha sido fornecido para *numTsmBuffers*. **dsmReleaseBuffer** também deve ser chamada se o aplicativo estiver prestes a chamar **dsmTerminate** e ainda mantiver buffers do IBM Spectrum Protect.

Sintaxe

```
dsInt16_t  dsmRequestBuffer   (getBufferIn_t        *dsmRequestBufferInP,
                               getBufferOut_t       *dsmRequestBufferOutP) ;
```

Parâmetros

getBufferIn_t * dsmRequestBufferInP (I)
Essa estrutura contém o seguinte parâmetro de entrada:

dsUint32_t dsmHandle
O identificador que identifica a sessão e a associa a uma chamada anterior de **dsmInitEx**.

getBufferOut_t *dsmRequestBufferOut P (0)
Essa estrutura contém os parâmetros de saída.

dsUint8_t tsmBufferHandle(0)
O identificador que identifica o buffer.

char *dataPtr(0)
O endereço no qual o aplicativo é gravado.

dsUint32_t *bufferLen(0)
O número máximo de bytes que pode ser gravado no buffer.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 47. Códigos de Retorno para dsmRequestBuffer

Código de retorno	Explicação
DSM_RC_BAD_CALL_SEQUENCE (33)	A chamada não foi emitida no estado apropriado.

Tabela 47. Códigos de Retorno para dsmRequestBuffer (continuação)

Código de retorno	Explicação
DSM_RC_SENDDATA_WITH_ZERO_SIZE (34)	Se o objeto que está sendo enviado tiver um comprimento igual a 0, nenhuma chamada para **dsmReleaseBuffer** será permitida.
DSM_RC_BUFF_ARRAY_ERROR (121)	Um buffer válido não pôde ser obtido.

dsmRetentionEvent

A chamada de função **dsmRetentionEvent** envia uma lista de IDs de objeto para o servidor IBM Spectrum Protect com uma operação de retenção de evento a ser executada por estes objetos. Utilize essa chamada de função nas chamadas de **dsmBeginTxn** e **dsmEndTxn**.

Nota: O servidor deve estar na versão 5.2.2.0 ou mais recente para que esta função funcione.

O número máximo de objetos de uma chamada está limitado ao valor de *maxObjPerTxn* que é retornado na estrutura *ApisessInfo* de uma chamada de **dsmQuerySessInfo**.

Somente um proprietário de um objeto pode enviar um evento nesse objeto.

Os seguintes eventos são possíveis:

eventRetentionActivate
Pode ser emitido somente para objetos ligados a um evento e baseado na classe de gerenciamento. O envio desse evento ativará o evento para o objeto e o estado da retenção para o objeto mudará de DSM_ARCH_RETINIT_PENDING para DSM_ARCH_RETINIT_STARTED.

eventHoldObj
Esse evento emitirá uma retenção ou exclusão suspensa no objeto de forma que, até uma liberação ser emitida, o objeto não expire e não possa ser excluído.

eventReleaseObj
Esse evento pode ser emitido somente para um objeto que tem um valor igual a DSM_ARCH_HELD_TRUE no campo ***objectHeld*** e removerá a suspensão do objeto retomando a política de retenção original.

Antes de enviar ***dsmRetentionEvent***, envie a seqüência de consulta descrita em "Consultando o sistema IBM Spectrum Protect" na página 34 para obter as informações para o objeto. A chamada para **dsmGetNextQObj** retorna uma estrutura de dados denominada **qryRespArchiveData** para consultas de archive. Essa estrutura de dados contém as informações necessárias para **dsmRetentionEvent**.

Sintaxe

```
extern dsInt16_t DSMLINKAGE   dsmRetentionEvent(
  dsmRetentionEventIn_t          *ddsmRetentionEventInP,
  dsmRetentionEventOut_t *dsmRetentionEventOutP
   );
```

Parâmetros

dsmRetentionEventIn_t *dsmRetentionEventP
Essa estrutura contém os seguintes parâmetros de entrada:

dsUint16_t stVersion;
Esse parâmetro indica a versão da estrutura.

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

dsmEventType_t evenType (I);
Esse parâmetro indica o tipo de evento. Consulte o início desta seção para obter o significado destes possíveis valores: **eventRetentionActivate**, **eventHoldObj**, **eventReleaseObj**

dsmObjList_t objList;
Esse parâmetro indica uma lista de IDs de objetos a sinalizar.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 48. Códigos de Retorno para dsmRetentionEvent

Código de retorno	Explicação
DSM_RC_ABORT_NODE_NOT_AUTHORIZED (36)	O nó ou o usuário não tem a autoridade apropriada.
DSM_RC_ABORT_TXN_LIMIT_EXCEEDED (249)	Número excessivo de objetos na transação.
DSM_RC_ABORT_OBJECT_ALREADY_HELD (250)	O objeto já está suspenso, não é possível emitir outra suspensão.
DSM_RC_REJECT_SERVER_DOWNLEVEL (58)	O servidor deve estar na versão V5.2.2.0 ou mais recente para que esta função funcione.

dsmSendBufferData

A chamada de função **dsmSendBufferData** envia um fluxo de bytes de dados para IBM Spectrum Protect por meio de um buffer que foi fornecido em uma chamada anterior de **dsmReleaseBuffer**. O cliente aplicativo pode transmitir qualquer tipo de dados para armazenamento no servidor. Geralmente, esses dados são dados do arquivo, mas não estão limitados aos dados do arquivo. Você pode chamar **dsmSendBufferData** várias vezes, se o fluxo de bytes de dados que você está enviando for grande. Independentemente de a chamada ser bem-sucedida ou falha, o buffer será liberado.

Restrição: Ao utilizar a opção *useTsmBuffers*, mesmo que um objeto seja incluído para compactação, ele não será compactado.

Sintaxe

```
dsInt16_t  dsmSendBufferData   (sendBufferDataIn_t        *dsmSendBufferDataExInP,
                                sendBufferDataOut_t       *dsmSendBufferDataOutP) ;
```

Parâmetros

sendBufferDataIn_t * dsmSendBufferDataInP (I)
Essa estrutura contém os seguintes parâmetros de entrada.

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEX** anterior.

dsUint8_t tsmBufferHandle(I)
O identificador que identifica o buffer a enviar.

char *dataPtr(I)
O endereço no qual os dados do aplicativo foram gravados.

dsUint32_t numBytes(I)
O número de bytes real gravado pelo aplicativo (deve sempre ser menos do que o valor fornecido em **dsmReleaseBuffer**).

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 49. Códigos de Retorno para dsmSendBufferData

Código de retorno	Explicação
DSM_RC_BAD_CALL_SEQUENCE (2041)	A chamada não foi emitida no estado apropriado.
DSM_RC_INVALID_TSMBUFFER (2042)	O identificador ou o valor de **dataPtr** é inválido.
DSM_RC_BUFF_ARRAY_ERROR (2045)	Ocorreu um erro de matriz do buffer.
DSM_RC_TOO_MANY_BYTES (2043)	O valor de ***numBytes*** é maior do que o tamanho do buffer fornecido na chamada de **dsmReleaseBuffer**.

dsmSendData

A chamada de função **dsmSendData** envia um fluxo de bytes de dados para o IBM Spectrum Protect por meio de um buffer. O cliente aplicativo pode transmitir qualquer tipo de dados para armazenamento no servidor. Geralmente, esses dados são dados do arquivo, mas não estão limitados aos dados do arquivo. Você pode chamar **dsmSendData** várias vezes, se o fluxo de bytes de dados que você deseja enviar for grande.

Restrição: O cliente aplicativo não pode reutilizar o buffer especificado em **dsmSendData** até a chamada de **dsmSendData** retornar.

Dica: Se IBM Spectrum Protect retornar o código 157 (DSM_RC_WILL_ABORT), inicie uma chamada para **dsmEndSendObj** e, em seguida, para **dsmEndTxn** com um voto igual a DSM_VOTE_COMMIT. O aplicativo, então, recebe o código de retorno 2302 (DSM_RC_CHECK_REASON_CODE) e transmite o código de razão novamente para o usuário do aplicativo. Isso informará ao usuário porque o servidor está encerrando a transação.

Sintaxe

```
dsInt16_t dsmSendData  (dsUint32_t       dsmHandle,
   DataBlk *dataBlkPtr);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

DataBlk *dataBlkPtr (I/O)
Esse parâmetro aponta para uma strutura que inclui um ponteiro para o buffer a partir do qual os dados devem ser enviados, assim como o tamanho do buffer. No retorno, essa estrutura contém o número de bytes que é realmente

transferido. Consulte o Apêndice B, “Arquivos de Origem de Definições de Tipos de API”, na página 163 para obter a definição do tipo.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 50. Códigos de Retorno para dsmSendData

Código de retorno	Explicação
DSM_RC_NO_COMPRESS_MEMORY (154)	Memória insuficente disponível para exceutar a compactação ou expansão de dados.
DSM_RC_COMPRESS_GREW (155)	Durante a compactação, os dados compactados aumentaram de tamanho em comparação aos dados originais.
DSM_RC_WILL_ABORT (157)	Ocorreu um erro desconhecido e inesperado, fazendo com que a transação fosse interrompida.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_NEEDTO_ENDTXN (2070)	É necessário encerrar a transação.
DSM_RC_OBJ_EXCLUDED (2080)	A lista incluir-excluir exclui o objeto.
DSM_RC_OBJ_NOBCG (2081)	O objeto não tem nenhum grupo de cópias de backup e não será enviado ao servidor.
DSM_RC_OBJ_NOACG (2082)	O objeto não tem nenhum grupo de cópias de archive e não será enviado para o servidor.
DSM_RC_SENDDATA_WITH_ZERO_SIZE (2107)	O objeto não pode enviar dados com um byte zero *sizeEstimate*.

dsmSendObj

A chamada de função **dsmSendObj** inicia um pedido para enviar um único objeto para o armazenamento. Várias chamadas de **dsmSendObj** e chamadas de **dsmSendData** associadas podem ser realizadas dentro dos limites de uma transação por motivos de desempenho.

A chamada de **dsmSendObj** processa os dados para o objeto como um fluxo de bytes transmitido em buffers de memória. O parâmetro **dataBlkPtr** na chamada de **dsmSendObj** permite que o cliente aplicativo execute um dos seguintes procedimentos:

- Transmita os dados e os atributos (os atributos são transmitidos através do **objAttrPtr**) do objeto em uma única chamada.
- Especifique parte dos dados do objeto atravésa da chamada de **dsmSendObj** e os demais dados através de uma ou mais chamadas de **dsmSendData**.

Como alternativa, o cliente aplicativo pode especificar somente os atributos através da chamada de **dsmSendObj** e especificar os dados de objetos através de uma ou mais chamadas a **dsmSendData**. Para esse método, configure **dataBlkPtr** para NULL na chamada de **dsmSendObj**.

Dica: Para determinados tipos de objetos, os dados do fluxo de bytes pode não ser associado aos dados; por exemplo, uma entrada de diretório sem nenhum atributo estendido.

Antes de **dsmSendObj** ser chamado, uma chamada anterior de **dsmBindMC** deve ser feita para ligar a classe de gerenciamento corretamente ao objeto do qual deseja fazer backup ou arquivar. A API mantém essa ligação para que possa associar a classe de gerenciamento apropriada ao objeto quando ele é enviado ao servidor. Se você permitir que a classe de gerenciamento ligada a uma chamada de **dsmSendObj** utilize por padrão um tipo de objeto igual a diretório (DSM_OBJ_DIRECTORY), o padrão pode não ser a classe de gerenciamento padrão. Em vez disso, a classe de gerenciamento com o maior tempo de retenção será utilizada. Se existir mais de uma classe de gerenciamento com esse tempo de retenção, a primeira encontrada será utilizada.

Siga todos os dados do objeto enviados para o armazenamento com uma chamada de **dsmEndSendObj**. Se você não tiver os dados de objeto a enviar ao servidor ou se todos os dados estavam contidos na chamada de **dsmSendObj**, inicie uma chamada de **dsmEndSendObj** antes de poder iniciar outra chamada de **dsmSendObj**. Se vários envios de dados tiverem sido requeridos através da chamada de **dsmSendData**, **dsmEndSendObj** segue o último envio para indicar a alteração de estado.

Dica: Se o IBM Spectrum Protect retornar o código 157 (DSM_RC_WILL_ABORT), inicie uma chamada para **dsmEndTxn** com um voto de DSM_VOTE_COMMIT. O aplicativo recebe o código de retorno 2302 (DSM_RC_CHECK_REASON_CODE) e transmite o código de razão novamente para o usuário do aplicativo. Isso informará ao usuário porque o servidor está encerrando a transação.

Se o código de razão for 11 (DSM_RS_ABORT_NO_REPOSIT_SPACE), é possível que *sizeEstimate* seja muito pequeno para a quantidade real de dados. O aplicativo precisa determinar um *sizeEstimate* mais preciso e enviar os dados novamente.

Sintaxe

```
dsInt16_t dsmSendObj  (dsUint32_t       dsmHandle,
   dsmSendType            sendType,
   void                   *sendBuff,
   dsmObjName             *objNameP,
   ObjAttr                *objAttrPtr,
   DataBlk *dataBlkPtr);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmSendType sendType (I)
Esse parâmetro especifica o tipo de envio que está sendo executado. Os valores possíveis são:

Nome	Descrição
stBackup	Um objeto de backup que é enviado ao servidor.
stArchive	Um objeto de archive que é enviado ao servidor.
stBackupMountWait	Um objeto de backup pelo qual deseja que o servidor aguarde até o dispositivo necessário, como uma fita, ser montado.
stArchiveMountWait	Um objeto de archive pelo qual deseja que o servidor aguarde até o dispositivo necessário, como uma fita, ser montado.

Nota: Utilize os tipos de **MountWait** se houver alguma possibilidade do usuário de aplicativo enviar dados para uma fita.

void *sendBuff (I)
Esse parâmetro é um ponteiro para uma estrutura que contém outras informações específicas para **sendType** na chamada. Atualmente, somente um **sendType** de **stArchive** tem uma estrutura associada. Essa estrutura é chamada **sndArchiveData** e contém a descrição de archive.

dsmObjName *objNameP (I)
Esse parâmetro é um ponteiro para a estrutura que contém o nome do espaço no arquivo, o nome do objeto de nível superior, o nome do objeto de nível inferior e o tipo de objeto. Consulte "IDs e Nomes de Objetos" na página 23 para obter mais informações.

ObjAttr *objAttrPtr (I)
Esse parâmetro transmite os atributos do objeto de interesse para o aplicativo. Consulte o Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163 para obter a definição do tipo.

Os atributos são:

- **owner** refere-se ao proprietário do objeto. Determinar se o proprietário está declarado como um nome específico ou uma cadeia vazia é importante ao obter o objeto novamente do armazenamento do IBM Spectrum Protect. Consulte "Acessando Objetos como Proprietário da Sessão" na página 25 para obter mais informações.
- **sizeEstimate** é a melhor estimativa de tamanho total do objeto de dados para enviar para o servidor. Seja o mais preciso possível para o tamanho, pois o servidor utiliza esse atributo para alocação de espaço eficiente e posicionamento do objeto em seus recursos de armazenamento.

  Se a estimativa de tamanho especificada for significativamente menor do que o número real de bytes enviados, o servidor pode ter dificuldades para alocar espaço suficiente e encerrar a transação com um código de razão igual a 11 (DSM_RS_ABORT_NO_REPOSIT_SPACE).

  Nota: A estimativa de tamanho é para o tamanho total do objeto de dados em bytes.

  Os objetos com um tamanho menor que DSM_MIN_COMPRESS_SIZE não são compactados.

  Se seu objeto não tiver dados de bits (somente as informações sobre o atributo da chamada), **sizeEstimate** deve ser zero.

  Nota: A partir da Versão 5.1.0, o destino da cópia em uma transação não é verificado para consistência em objetos de comprimento zero.
- **objCompressed** é um valor booleano que indica se os dados do objeto já foram compactados ou não.

  Se o objeto estiver compactado (objeto *compressed=bTrue*), o IBM Spectrum Protect não tentará compactá-lo novamente. Se ele não estiver compactado, o IBM Spectrum Protect decidirá se deve compactar o objeto, com base nos valores da opção `compression` configurada pelo administrador e configurada nas fontes de configuração da API.

  Se seu aplicativo planeja utilizar restauração ou recuperação parcial do objeto, não é possível compactar os dados durante o envio. Para aplicar isso, configure *ObjAttr.objCompressed* como *bTrue*.
- **objInfo** salva informações sobre o objeto específico.

Restrição: As informações não são armazenadas aqui automaticamente. Quando esse atributo é usado, deve-se configurar o atributo *objInfoLength*, para mostrar o comprimento de *objInfo*.

- **mcNameP** contém o nome de uma classe de gerenciamento que substitui a classe de gerenciamento obtida de **dsmBindMC**.
- **disableDeduplication** é um valor Booleano. Quando ele é definido como true, este objeto não é deduplicado pelo cliente.

DataBlk *dataBlkPtr (I/O)

Esse parâmetro aponta para uma estrutura que inclui um ponteiro para o buffer de dados do qual deve-se fazer backup ou archive e o tamanho do buffer. Esse parâmetro aplica-se somente a **dsmSendObj**. Se quiser começar a enviar dados em uma chamada subseqüente de **dsmSendData**, e não na chamada de **dsmSendObj**, configure o ponteiro do buffer na estrutura DataBlk para NULL. No retorno, essa estrutura contém o número de bytes que é realmente transferido. Consulte o Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163 para obter a definição do tipo.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 51. Códigos de Retorno para dsmSendObj

Código de retorno	Explicação
DSM_RC_NO_COMPRESS_MEMORY (154)	Memória insuficente disponível para exceutar a compactação ou expansão de dados.
DSM_RC_COMPRESS_GREW (155)	Durante a compactação, os dados compactados aumentaram de tamanho em comparação aos dados originais.
DSM_RC_WILL_ABORT (157)	Ocorreu um erro desconhecido e inesperado, fazendo com que a transação fosse interrompida.
DSM_RC_TL_NOACG (186)	A classe de gerenciamento desse arquivo não tem um grupo de cópias válido para o tipo de envio.
DSM_RC_NULL_OBJNAME (2000)	O nome do objeto é nulo.
DSM_RC_NULL_OBJATTRPTR (2004)	O ponteiro do atributo do objeto é nulo.
DSM_RC_INVALID_OBJTYPE (2010)	O tipo de objeto é inválido.
DSM_RC_INVALID_OBJOWNER (2019)	O proprietário do objeto é inválido.
DSM_RC_INVALID_SENDTYPE (2022)	Tipo de envio inválido.
DSM_RC_WILDCHAR_NOTALLOWED (2050)	Caracteres curinga não são permitidos.
DSM_RC_FS_NOT_REGISTERED (2061)	O espaço no arquivo não está registrado.
DSM_RC_WRONG_VERSION_PARM (2065)	A versão da API do cliente aplicativo é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_NEEDTO_ENDTXN (2070)	É necessário encerrar a transação.
DSM_RC_OBJ_EXCLUDED (2080)	A lista incluir-excluir excluiu o objeto.
DSM_RC_OBJ_NOBCG (2081)	O objeto não tem nenhum grupo de cópias de backup e não será enviado ao servidor.
DSM_RC_OBJ_NOACG (2082)	O objeto não tem nenhum grupo de cópias de archive e não será enviado ao servidor.
DSM_RC_DESC_TOOLONG (2100)	A descrição é muito longa.
DSM_RC_OBJINFO_TOOLONG (2101)	As informações do objeto são muito longas.

Tabela 51. Códigos de Retorno para dsmSendObj (continuação)

Código de retorno	Explicação
DSM_RC_HL_TOOLONG (2102)	O qualificador de nível superior é muito longo.
DSM_RC_FILESPACE_TOOLONG (2104)	O nome do espaço no arquivo é muito longo.
DSM_RC_LL_TOOLONG (2105)	O qualificador de nível inferior é muito longo.
DSM_RC_NEEDTO_CALL_BINDMC (2301)	**dsmBindMC** deve ser chamado primeiro.

dsmSetAccess

A chamada de função **dsmSetAccess** fornece a outros usuários ou nós acesso a versões de backup ou cópias arquivadas de seus objetos, acesso a todos os seus objetos ou acesso a um conjunto seletivo. Quando você concede acesso a outro usuário, esse usuário pode consultar, restaurar ou recuperar seus arquivos. Esse comando suporta curingas para os seguintes campos: *fs*, *hl*, *ll*, *node*, *owner*.

Nota: Você não pode conceder acesso a versões de backup e a cópias de archive, utilizando um único comando. Você deve especificar backup ou archive.

Sintaxe

```
dsInt16_t DSMLINKAGE dsmSetAccess
        (dsUint32_t          dsmHandle,
         dsmSetAccessType    accessType,
         dsmObjName             *objNameP,
         char                   *node,
         char                *owner);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmAccessType accessType (I)
Esse parâmetro especifica o tipo de objetos para o qual deseja fornecer acesso. Os valores possíveis são:

Nome	Descrição
atBackup	Especifica que o acesso está sendo configurado para fazer backup de objetos.
atArchive	Especifica que o acesso está sendo configurado para objetos de archive.

dsmObjName *objNameP (I)
Esse parâmetro é um ponteiro para a estrutura que contém o nome do espaço no arquivo, o nome do objeto de nível superior e o nome do objeto de nível inferior.

Nota: Para especificar todos os espaços no arquivo, utilize um asterisco (*) para o nome do espaço no arquivo.

char *node (I)
Esse parâmetro é um ponteiro para o nome do nó o qual é concedido acesso. Para qualquer nó, especifique um asterisco (*).

char *owner (I)
Esse parâmetro é um ponteiro para o nome do usuário no nó para o qual você concedeu acesso. Para todos os usuários, especifique um asterisco (*).

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 52. Códigos de Retorno para dsmSetAccess

Código de retorno	Explicação
DSM_RC_INVALID_ACCESS_TYPE (2110)	O tipo de acesso especificado é inválido.
DSM_RC_FILE_SPACE_NOT_FOUND (124)	O espaço no arquivo especificado não foi localizado no servidor.
DSM_RC_QUERY_COMM_FAILURE (2111)	Erro de comunicação durante a consulta do servidor.
DSM_RC_NO_FILES_BACKUP (2112)	Não foi feito backup de nenhum arquivo para esse espaço no arquivo.
DSM_RC_NO_FILES_ARCHIVE (2113)	Nenhum arquivo foi arquivado para espaço no arquivo.
DSM_RC_INVALID_SETACCESS (2114)	Formulação inválida de acesso configurado.

dsmSetUp

A chamada da função **dsmSetUp** sobrescreve os valores das variáveis de ambiente. Chame **dsmSetUp**antes de **dsmInitEx**. Os valores transmitidos na estrutura **envSetUp** sobrescreverão quaisquer variáveis de ambiente ou padrões existentes. Se você especificar NULL para um campo, os valores serão retirados do ambiente. Se você não configurar um valor, os valores serão retirados dos padrões.

Requisitos:

1. Se você utilizar **dsmSetUp**, sempre chame **dsmTerminate**antes de**dsmCleanUp**.
2. A instrumentação da API só poderá ser ativada se a API testflag INSTRUMENT: estiver configurada no arquivo de configuração e as chamadas **dsmSetUp** ou **dsmCleanUp** forem usadas no aplicativo.

Sintaxe

```
dsInt16_t DSMLINKAGE dsmSetUp
        (dsBool_t    mtFlag,
         envSetUp    *envSetUpP);
```

Parâmetros

dsBool_t mtFlag (I)
Esse parâmetro especifica que a API foi utilizada no modo de encadeamento único ou multiencadeado. Os valores incluem:

```
DSM_SINGLETHREAD
DSM_MULTITHREAD
```

Exigência: O sinalizador de multiencadeamento deve estar ativado para que a transferência de dados sem a LAN ocorra.

envSetUp *envSetUpP(I)
Esse parâmetro é um ponteiro para a estrutura que contém os valores para sobrescrever. Especifique NULL se não quiser sobrescrever as variáveis de ambiente existentes. Os campos na estrutura **envSetUp** incluem:

Nome	Descrição
dsmiDir	Um caminho de diretório completo que contém um arquivo de mensagem no UNIX ou Linux. Também especifica os diretórios dsmtca e dsm.sys.
dsmiConfig	O nome completo do arquivo de opções do cliente.
dsmiLog	O caminho completo do diretório de log de erros.
argv	Transmita o nome argv[0] do programa de chamada caso o aplicativo deva ser executado com autoridade de usuário autorizado. Consulte “Configurando a Opção `passwordaccess` para `generate` Sem TCA” na página 21 para obter mais informações.
logName	O nome do arquivo de um log de erros se o aplicativo não utilizar dsierror.log.
inclExclCaseSensitive	Indica se as regras de inclusão/exclusão fazem ou não distinção entre maiúsculas e minúsculas. Esse parâmetro pode ser utilizado somente no Windows, ele será ignorado em outros.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 53. Códigos de Retorno para dsmSetUp

Código de retorno	Explicação
DSM_RC_ACCESS_DENIED (106)	Acesso ao arquivo ou diretório especificado negado.
DSM_RC_INVALID_OPT (0400)	Uma opção inválida foi localizada.
DSM_RC_NO_HOST_ADDR (0405)	O TCPSERVERADDRESS desse servidor não está definido na sub-rotina do nome do servidor no arquivo de opções do sistema.
DSM_RC_NO_OPT_FILE (0406)	O arquivo de opções especificado por nome do arquivo não pode ser localizado.
DSM_RC_MACHINE_SAME (0408)	NODENAME definido no arquivo de opções não pode ser o mesmo que o *`HostName` do sistema*.
DSM_RC_INVALID_SERVER (0409)	O arquivo de opções do sistema não contém a opção SERVERNAME.
DSM_RC_INVALID_KEYWORD (0410)	Uma palavra-chave de opção inválida foi localizada no arquivo de configuração **dsmInitEx**, a cadeia de opções, dsm.sys ou dsm.opt.
DSM_RC_PATTERN_TOO_COMPLEX (0411)	O padrão de inclusão ou exclusão emitido é muito complexo para ser interpretado precisamente pelo IBM Spectrum Protect.
DSM_RC_NO_CLOSING_BRACKET (0412)	O padrão de inclusão ou exclusão foi construído incorretamente. O colchete direito está faltando.
DSM_RC_NLS_CANT_OPEN_TXT (0610)	O sistema não pode abrir o arquivo de texto de mensagem.
DSM_RC_NLS_INVALID_CNTL_REC (0612)	O sistema não pode utilizar o arquivo de texto de mensagem.
DSM_RC_NOT_ADSM_AUTHORIZED (0927)	Você deve ser o usuário autorizado para ter multiencadeamento e geração de *`passwordaccess`*.
DSM_RC_NO_INCLEXCL_FILE (2229)	O arquivo include-exclude não foi localizado.
DSM_RC_NO_SYS_OR_INCLEXCL (2230)	O arquivo dsm.sys ou include-exclude não foi localizado.

dsmTerminate

A chamada de função **dsmTerminate** encerra uma sessão com o servidor IBM Spectrum Protect e limpa o ambiente do IBM Spectrum Protect.

Sintaxe

Não há códigos de retorno específicos para essa chamada.

```
dsInt16_t dsmTerminate  (dsUint32_t dsmHandle);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmUpdateFS

A chamada de função **dsmUpdateFS** atualiza um espaço no arquivo no armazenamento do IBM Spectrum Protect. Essa atualização assegura que o administrador tenha um registro atual de seu espaço no arquivo.

Sintaxe

```
dsInt16_t dsmUpdateFS  (dsUint32_t       dsmHandle,
   char                   *fs,
   dsmFSUpd               *fsUpdP,
   dsUint32_t      fsUpdAct);
```

Parâmetros

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

char *fs (I)
Esse parâmetro é um ponteiro para o nome do espaço no arquivo.

dsmFSUpd *fsUpdP (I)
Este parâmetro é um ponteiro para a estrutura que possui os campos corretos para a atualização que você deseja. Preencha somente os campos que precisam de atualização.

dsUint32_t fsUpdAct (I)
Um bitmap de 2 bytes que indica qual dos campos deve ser atualizado. As máscaras de bits possuem os valores a seguir:

- DSM_FSUPD_FSTYPE
- DSM_FSUPD_FSINFO

  Dica: Para sistemas operacionais Windows, o valor da letra de unidade de **dsmDOSAttrib** também é atualizado quando **FSINFO** é selecionado.
- DSM_FSUPD_OCCUPANCY
- DSM_FSUPD_CAPACITY
- DSM_FSUPD_BACKSTARTDATE
- DSM_FSUPD_BACKCOMPLETEDATE

Para obter uma descrição destas máscaras de bits, consulte as definições de DSM_FSUPD no tópico a seguir: Apêndice B, "Arquivos de Origem de Definições de Tipos de API", na página 163.

Códigos de retorno

A tabela a seguir lista códigos de retorno para a chamada de função **dsmUpdateFS**.

Tabela 54. Códigos de Retorno para dsmUpdateFS

Código de retorno	Número do código de retorno	Descrição
DSM_RC_FS_NOT_REGISTERED	2061	O nome do espaço no arquivo não está registrado.
DSM_RC_WRONG_VERSION_PARM	2065	A versão de API do aplicativo cliente é diferente da versão da biblioteca do IBM Spectrum Protect.
DSM_RC_FSINFO_TOOLONG	2106	As informações do espaço no arquivo são muito longas.

dsmUpdateObj

A chamada de função **dsmUpdateObj** atualiza as meta informações associadas a um objeto de backup ou archive ativo que já se encontra no servidor. Os dados de bit do aplicativo não serão afetados. Para atualizar um objeto, você deve fornecer um nome específico sem curinga. Para atualizar um objeto com archive, configure **dsmSendType** para **stArchive**. Somente o objeto de archive denominado por último é atualizado.

A chamada de **dsmUpdateObj** pode ser iniciada somente no estado de sessão; não pode ser chamada em uma transação, pois executa sua própria transação. E você pode atualizar somente um objeto por vez.

Restrição: Em um sistema operacional UNIX ou Linux, se você alterar o campo do proprietário, não será possível consultar ou restaurar o objeto a não ser que você seja o usuário root.

Sintaxe

```
dsInt16_t  dsmUpdateObj
   (dsUint32_t          dsmHandle,
    dsmSendType           sendType,
    void                   *sendBuff,
    dsmObjName            *objNameP,
    ObjAttr        *objAttrPtr,    /* objInfo */
    dsUint16_t        objUpdAct);    /* vetor de bit de ação */
```

Parâmetros

As descrições dos campos são as mesmas que as de **dsmSendObj**, com as seguintes exceções:

dsmObjName *objNameP (I)
Você não pode utilizar um curinga.

ObjAttr *objAttrPtr (I)
O campo **objCompressed** é ignorado para essa chamada.

Outras diferenças são:

- **owner**. Se você especificar um novo campo **owner**, o proprietário será alterado.

- **sizeEstimate**. Se você especificar um valor diferente de zero, ele será a quantidade real de dados enviados, em bytes. O valor será armazenado nos metadados do IBM Spectrum Protect para uso futuro.
- **objInfo**. Esse atributo contém as novas informações a serem colocadas no campo **objInfo**. Configure **objInfoLength** para o comprimento do novo **obiInfo**.

dsUint16_t objUpdAct
As máscaras de bits e possíveis ações para **objUpdAct** são:

DSM_BACKUPD_MC
Atualiza a classe de gerenciamento para o objeto.

DSM_BACKUPD_OBJINFO
Atualiza **objInfo**, **objInfoLength** e **sizeEstimate**.

DSM_BACKUPD_OWNER
Atualiza o proprietário do objeto.

DSM_ARCHUPD_DESCR
Atualiza o campo **Description**. Digite o valor para a nova descrição através do parâmetro **SendBuff**. Consulte o programa de amostra para utilização apropriada.

DSM_ARCHUPD_OBJINFO
Atualiza **objInfo**, **objInfoLength** e **sizeEstimate**.

DSM_ARCHUPD_OWNER
Atualiza o proprietário do objeto.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ).

Tabela 55. Códigos de Retorno para dsmUpdateObj

Código de retorno	Explicação
DSM_RC_INVALID_ACTION (2232)	Ação inválida.
DSM_RC_FS_NOT_REGISTERED (2061)	O espaço no arquivo não está registrado.
DSM_RC_BAD_CALL_SEQUENCE (2041)	A seqüência de chamadas é inválida.
DSM_RC_WILDCHAR_NOTALLOWED (2050)	Caracteres curinga não são permitidos.
DSM_RC_ABORT_NO_MATCH (2)	A consulta anterior não corresponde.

dsmUpdateObjEx

A chamada de função **dsmUpdateObjEx** atualiza as meta informações associadas a um objeto de backup ou de archive ativo que está no servidor. Os dados de bit do aplicativo não serão afetados. Para atualizar um objeto, você deve especificar um nome sem curinga ou especificar o ID de objeto para atualizar um objeto arquivado específico. Não é possível utilizar caracteres curinga ao especificar o nome. Para atualizar um objeto de backup, configure o parâmetro **dsmSendType** para **stBackup**. Para atualizar um objeto arquivado, configure o parâmetro **dsmSendType** para **stArchive**.

A chamada **dsmUpdateObjEx** pode ser iniciada apenas no estado de sessão; ela não pode ser chamada dentro de uma transação porque desempenha sua própria transação. É possível atualizar apenas um objeto por vez.

Restrição: Em um sistema operacional UNIX ou Linux, se você alterar o campo do proprietário, não será possível consultar ou restaurar o objeto a não ser que você seja o usuário root. Apenas a versão ativa atual de um objeto de backup pode ser atualizada.

Sintaxe

```
dsInt16_t  dsmUpdateObjEx
   (dsmUpdateObjExIn_t   *dsmUpdateObjExInP,
    dsmUpdateObjExOut_t  *dsmUpdateObjExOutP);
```

Parâmetros

dsmUpdateObjExIn_t *dsmUpdateObjExInP
Essa estrutura contém os seguintes parâmetros de entrada:

dsUint16_t stVersion (I)
A versão atual da estrutura que é utilizada.

dsUint32_t dsmHandle (I)
O identificador que associa essa chamada a uma chamada de **dsmInitEx** anterior.

dsmSendType sendType (I)
O tipo de envio que está sendo desempenhado. O valor pode ser:

stBackup
Um objeto de backup que é enviado ao servidor.

stArchive
Um objeto de archive que é enviado ao servidor.

dsmObjName *objNameP (I)
Um ponteiro para a estrutura que contém o nome do espaço no arquivo, o nome do objeto de alto nível, o nome do objeto de nível inferior e o tipo de objeto. Você não pode utilizar um curinga.

ObjAttr *objAttrPtr (I)
Transmite atributos do objeto ao aplicativo. Os valores que são atualizados dependem dos sinalizadores no campo **objUpdAct**. O atributo **objCompressed** é ignorado para esta chamada.

Os atributos são:

- **owner** altera o proprietário se um novo nome for digitado.
- **sizeEstimate** é a quantidade real de dados enviados em bytes. O valor será armazenado nos metadados do IBM Spectrum Protect para uso futuro.
- **objCompressed** é um valor booleano que indica se os dados do objeto já foram compactados ou não.
- **objInfo** é um atributo que contém as novas informações a serem colocadas no campo **objInfo**. Configure **objInfoLength** para o comprimento do novo **objInfo**.
- **mcNameP** contém o nome de uma classe de gerenciamento que substitui a classe de gerenciamento obtida de **dsmBindMC**.

dsUint32_t objUpdAct
Especifica as máscaras de bits e as ações para **objUpdAct**:

DSM_BACKUPD_MC
Atualiza a classe de gerenciamento para o objeto.

DSM_BACKUPD_OBJINFO
Atualiza o objeto de informações (**objInfo**), o comprimento do objeto de informações (**objInfoLength**) e a quantidade de dados enviados (**sizeEstimate**) para o objeto de backup.

DSM_BACKUPD_OWNER
Atualiza o proprietário para o objeto de backup.

DSM_ARCHUPD_DESCR
Atualiza o campo **Descrição** para o objeto de archive. Digite o valor para a nova descrição através do parâmetro **sendBuff**.

DSM_ARCHUPD_OBJINFO
Atualiza o objeto de informações (**objInfo**), o comprimento do objeto de informações (**objInfoLength**) e a quantidade de dados enviados (**sizeEstimate**) para o objeto de archive.

DSM_ARCHUPD_OWNER
Atualiza o proprietário do objeto de archive.

ObjID archObjId
Especifica o ID de objeto exclusivo para um objeto de archive específico. Como vários objetos de archive podem ter o mesmo nome, este parâmetro identifica um específico. O ID de objeto pode ser obtido utilizando-se uma chamada de archive de consulta.

dsmUpdateObjExOut_t *dsmUpdateObjExOutP
Esta estrutura contém o parâmetro de saída:

dsUint16_t stVersion (I)
A versão atual da estrutura que é utilizada.

Códigos de retorno

Os números do código de retorno são fornecidos entre parênteses ( ) na tabela a seguir.

Tabela 56. Códigos de Retorno para dsmUpdateObjEx

Código de retorno	Explicação
DSM_RC_INVALID_ACTION (2012)	Ação inválida.
DSM_RC_FS_NOT_REGISTERED (2061)	O espaço no arquivo não está registrado.
DSM_RC_BAD_CALL_SEQUENCE (2041)	A sequência de chamadas é inválida.
DSM_RC_WILDCHAR_NOTALLOWED (2050)	Caracteres curinga não são permitidos.
DSM_RC_ABORT_NO_MATCH (2)	A consulta anterior não corresponde.

Apêndice A. Arquivo de Origem dos Códigos de Retorno do API : dsmrc.h

O arquivo de cabeçalho dsmrc.h contém todos os códigos de retorno que o API pode retornar para um aplicativo.

As informações fornecidas aqui contêm uma cópia point-in-time do arquivo dsmrc.h que é distribuído com a API. Visualize o arquivo no pacote de distribuição da API para a versão mais recente.

```
/*****************************************************************
* Tivoli Storage Manager                                         *
* Componente do Cliente da API                                   *
*                                                                *
* (C) Copyright IBM Corporation 1993,2010                        *
*****************************************************************/

/*****************************************************************/
/* Nome do Arquivo de Cabeçalho: dsmrc.h                         */
/*                                                               */
/* Nome descritivo: Códigos de retorno das APIs do Tivoli Storage */
/* Manager                                                       */
/*****************************************************************/
#ifndef _H_DSMRC
#define _H_DSMRC


#ifndef DSMAPILIB

#ifndef _H_ANSMACH
typedef int RetCode ;
#endif

#endif


#define DSM_RC_SUCCESSFUL               0 /* conclusão bem-sucedida  */
#define DSM_RC_OK                       0 /* conclusão bem-sucedida  */

#define DSM_RC_UNSUCCESSFUL            -1 /* conclusão mal sucedida  */

/* código de razão dsmEndTxn */
#define DSM_RS_ABORT_SYSTEM_ERROR           1
#define DSM_RS_ABORT_NO_MATCH               2
#define DSM_RS_ABORT_BY_CLIENT              3
#define DSM_RS_ABORT_ACTIVE_NOT_FOUND       4
#define DSM_RS_ABORT_NO_DATA                5
#define DSM_RS_ABORT_BAD_VERIFIER           6
#define DSM_RS_ABORT_NODE_IN_USE            7
#define DSM_RS_ABORT_EXPDATE_TOO_LOW        8
#define DSM_RS_ABORT_DATA_OFFLINE           9
#define DSM_RS_ABORT_EXCLUDED_BY_SIZE       10
#define DSM_RS_ABORT_NO_STO_SPACE_SKIP      11
#define DSM_RS_ABORT_NO_REPOSIT_SPACE       DSM_RS_ABORT_NO_STO_SPACE_SKIP
#define DSM_RS_ABORT_MOUNT_NOT_POSSIBLE     12
#define DSM_RS_ABORT_SIZESTIMATE_EXCEED     13
#define DSM_RS_ABORT_DATA_UNAVAILABLE       14
#define DSM_RS_ABORT_RETRY                  15
#define DSM_RS_ABORT_NO_LOG_SPACE           16
#define DSM_RS_ABORT_NO_DB_SPACE            17
#define DSM_RS_ABORT_NO_MEMORY              18

#define DSM_RS_ABORT_FS_NOT_DEFINED         20
#define DSM_RS_ABORT_NODE_ALREADY_DEFED     21
#define DSM_RS_ABORT_NO_DEFAULT_DOMAIN      22
#define DSM_RS_ABORT_INVALID_NODENAME       23
#define DSM_RS_ABORT_INVALID_POL_BIND       24
#define DSM_RS_ABORT_DEST_NOT_DEFINED       25
#define DSM_RS_ABORT_WAIT_FOR_SPACE         26
#define DSM_RS_ABORT_NOT_AUTHORIZED         27
#define DSM_RS_ABORT_RULE_ALREADY_DEFED     28
```

```
#define DSM_RS_ABORT_NO_STOR_SPACE_STOP      29

#define DSM_RS_ABORT_LICENSE_VIOLATION       30
#define DSM_RS_ABORT_EXTOBJID_ALREADY_EXISTS 31
#define DSM_RS_ABORT_DUPLICATE_OBJECT        32

#define DSM_RS_ABORT_INVALID_OFFSET          33    /* Recuperação Parcial do Objeto */
#define DSM_RS_ABORT_INVALID_LENGTH          34    /* Recuperação Parcial do Objeto */
#define DSM_RS_ABORT_STRING_ERROR            35
#define DSM_RS_ABORT_NODE_NOT_AUTHORIZED     36
#define DSM_RS_ABORT_RESTART_NOT_POSSIBLE    37
#define DSM_RS_ABORT_RESTORE_IN_PROGRESS     38
#define DSM_RS_ABORT_SYNTAX_ERROR            39

#define DSM_RS_ABORT_DATA_SKIPPED            40
#define DSM_RS_ABORT_EXCEED_MAX_MP           41
#define DSM_RS_ABORT_NO_OBJSET_MATCH         42
#define DSM_RS_ABORT_PVR_ERROR               43
#define DSM_RS_ABORT_BAD_RECOGTOKEN          44
#define DSM_RS_ABORT_MERGE_ERROR             45
#define DSM_RS_ABORT_FSRENAME_ERROR          46
#define DSM_RS_ABORT_INVALID_OPERATION       47
#define DSM_RS_ABORT_STGPOOL_UNDEFINED       48
#define DSM_RS_ABORT_INVALID_DATA_FORMAT     49
#define DSM_RS_ABORT_DATAMOVER_UNDEFINED     50

#define DSM_RS_ABORT_INVALID_MOVER_TYPE      231
#define DSM_RS_ABORT_ITEM_IN_USE             232
#define DSM_RS_ABORT_LOCK_CONFLICT           233
#define DSM_RS_ABORT_SRV_PLUGIN_COMM_ERROR   234
#define DSM_RS_ABORT_SRV_PLUGIN_OS_ERROR     235
#define DSM_RS_ABORT_CRC_FAILED              236
#define DSM_RS_ABORT_INVALID_GROUP_ACTION    237
#define DSM_RS_ABORT_DISK_UNDEFINED          238
#define DSM_RS_ABORT_BAD_DESTINATION         239
#define DSM_RS_ABORT_DATAMOVER_NOT_AVAILABLE 240
#define DSM_RS_ABORT_STGPOOL_COPY_CONT_NO    241
#define DSM_RS_ABORT_RETRY_SINGLE_TXN        242
#define DSM_RS_ABORT_TOC_CREATION_FAIL       243
#define DSM_RS_ABORT_TOC_LOAD_FAIL           244
#define DSM_RS_ABORT_PATH_RESTRICTED         245
#define DSM_RS_ABORT_NO_LANFREE_SCRATCH      246
#define DSM_RS_ABORT_INSERT_NOT_ALLOWED      247
#define DSM_RS_ABORT_DELETE_NOT_ALLOWED      248
#define DSM_RS_ABORT_TXN_LIMIT_EXCEEDED      249
#define DSM_RS_ABORT_OBJECT_ALREADY_HELD     250
#define DSM_RS_ABORT_INVALID_CHUNK_REFERENCE 254
#define DSM_RS_ABORT_DESTINATION_NOT_DEDUP   255
#define DSM_RS_ABORT_DESTINATION_POOL_CHANGED 257
#define DSM_RS_ABORT_NOT_ROOT                258


/* CÓDIGO DE RETORNO */

#define DSM_RC_ABORT_SYSTEM_ERROR            DSM_RS_ABORT_SYSTEM_ERROR
#define DSM_RC_ABORT_NO_MATCH                DSM_RS_ABORT_NO_MATCH
#define DSM_RC_ABORT_BY_CLIENT               DSM_RS_ABORT_BY_CLIENT
#define DSM_RC_ABORT_ACTIVE_NOT_FOUND        DSM_RS_ABORT_ACTIVE_NOT_FOUND
#define DSM_RC_ABORT_NO_DATA                 DSM_RS_ABORT_NO_DATA
#define DSM_RC_ABORT_BAD_VERIFIER            DSM_RS_ABORT_BAD_VERIFIER
#define DSM_RC_ABORT_NODE_IN_USE             DSM_RS_ABORT_NODE_IN_USE
#define DSM_RC_ABORT_EXPDATE_TOO_LOW         DSM_RS_ABORT_EXPDATE_TOO_LOW
#define DSM_RC_ABORT_DATA_OFFLINE            DSM_RS_ABORT_DATA_OFFLINE
#define DSM_RC_ABORT_EXCLUDED_BY_SIZE        DSM_RS_ABORT_EXCLUDED_BY_SIZE

#define DSM_RC_ABORT_NO_REPOSIT_SPACE        DSM_RS_ABORT_NO_STO_SPACE_SKIP
#define DSM_RC_ABORT_NO_STO_SPACE_SKIP       DSM_RS_ABORT_NO_STO_SPACE_SKIP

#define DSM_RC_ABORT_MOUNT_NOT_POSSIBLE      DSM_RS_ABORT_MOUNT_NOT_POSSIBLE
#define DSM_RC_ABORT_SIZESTIMATE_EXCEED      DSM_RS_ABORT_SIZESTIMATE_EXCEED
#define DSM_RC_ABORT_DATA_UNAVAILABLE        DSM_RS_ABORT_DATA_UNAVAILABLE
#define DSM_RC_ABORT_RETRY                   DSM_RS_ABORT_RETRY
#define DSM_RC_ABORT_NO_LOG_SPACE            DSM_RS_ABORT_NO_LOG_SPACE
#define DSM_RC_ABORT_NO_DB_SPACE             DSM_RS_ABORT_NO_DB_SPACE
#define DSM_RC_ABORT_NO_MEMORY               DSM_RS_ABORT_NO_MEMORY

#define DSM_RC_ABORT_FS_NOT_DEFINED          DSM_RS_ABORT_FS_NOT_DEFINED
#define DSM_RC_ABORT_NODE_ALREADY_DEFED      DSM_RS_ABORT_NODE_ALREADY_DEFED
#define DSM_RC_ABORT_NO_DEFAULT_DOMAIN       DSM_RS_ABORT_NO_DEFAULT_DOMAIN
```

```
#define DSM_RC_ABORT_INVALID_NODENAME        DSM_RS_ABORT_INVALID_NODENAME
#define DSM_RC_ABORT_INVALID_POL_BIND        DSM_RS_ABORT_INVALID_POL_BIND
#define DSM_RC_ABORT_DEST_NOT_DEFINED        DSM_RS_ABORT_DEST_NOT_DEFINED
#define DSM_RC_ABORT_WAIT_FOR_SPACE          DSM_RS_ABORT_WAIT_FOR_SPACE
#define DSM_RC_ABORT_NOT_AUTHORIZED          DSM_RS_ABORT_NOT_AUTHORIZED
#define DSM_RC_ABORT_RULE_ALREADY_DEFED      DSM_RS_ABORT_RULE_ALREADY_DEFED
#define DSM_RC_ABORT_NO_STOR_SPACE_STOP      DSM_RS_ABORT_NO_STOR_SPACE_STOP

#define DSM_RC_ABORT_LICENSE_VIOLATION       DSM_RS_ABORT_LICENSE_VIOLATION
#define DSM_RC_ABORT_EXTOBJID_ALREADY_EXISTS DSM_RS_ABORT_EXTOBJID_ALREADY_EXISTS
#define DSM_RC_ABORT_DUPLICATE_OBJECT        DSM_RS_ABORT_DUPLICATE_OBJECT

#define DSM_RC_ABORT_INVALID_OFFSET          DSM_RS_ABORT_INVALID_OFFSET
#define DSM_RC_ABORT_INVALID_LENGTH          DSM_RS_ABORT_INVALID_LENGTH

#define DSM_RC_ABORT_STRING_ERROR            DSM_RS_ABORT_STRING_ERROR
#define DSM_RC_ABORT_NODE_NOT_AUTHORIZED     DSM_RS_ABORT_NODE_NOT_AUTHORIZED
#define DSM_RC_ABORT_RESTART_NOT_POSSIBLE    DSM_RS_ABORT_RESTART_NOT_POSSIBLE
#define DSM_RC_ABORT_RESTORE_IN_PROGRESS     DSM_RS_ABORT_RESTORE_IN_PROGRESS
#define DSM_RC_ABORT_SYNTAX_ERROR            DSM_RS_ABORT_SYNTAX_ERROR

#define DSM_RC_ABORT_DATA_SKIPPED            DSM_RS_ABORT_DATA_SKIPPED
#define DSM_RC_ABORT_EXCEED_MAX_MP           DSM_RS_ABORT_EXCEED_MAX_MP
#define DSM_RC_ABORT_NO_OBJSET_MATCH         DSM_RS_ABORT_NO_OBJSET_MATCH
#define DSM_RC_ABORT_PVR_ERROR               DSM_RS_ABORT_PVR_ERROR
#define DSM_RC_ABORT_BAD_RECOGTOKEN          DSM_RS_ABORT_BAD_RECOGTOKEN
#define DSM_RC_ABORT_MERGE_ERROR             DSM_RS_ABORT_MERGE_ERROR
#define DSM_RC_ABORT_FSRENAME_ERROR          DSM_RS_ABORT_FSRENAME_ERROR
#define DSM_RC_ABORT_INVALID_OPERATION       DSM_RS_ABORT_INVALID_OPERATION
#define DSM_RC_ABORT_STGPOOL_UNDEFINED       DSM_RS_ABORT_STGPOOL_UNDEFINED
#define DSM_RC_ABORT_INVALID_DATA_FORMAT     DSM_RS_ABORT_INVALID_DATA_FORMAT
#define DSM_RC_ABORT_DATAMOVER_UNDEFINED     DSM_RS_ABORT_DATAMOVER_UNDEFINED

#define DSM_RC_ABORT_INVALID_MOVER_TYPE      DSM_RS_ABORT_INVALID_MOVER_TYPE
#define DSM_RC_ABORT_ITEM_IN_USE             DSM_RS_ABORT_ITEM_IN_USE
#define DSM_RC_ABORT_LOCK_CONFLICT           DSM_RS_ABORT_LOCK_CONFLICT
#define DSM_RC_ABORT_SRV_PLUGIN_COMM_ERROR   DSM_RS_ABORT_SRV_PLUGIN_COMM_ERROR
#define DSM_RC_ABORT_SRV_PLUGIN_OS_ERROR     DSM_RS_ABORT_SRV_PLUGIN_OS_ERROR
#define DSM_RC_ABORT_CRC_FAILED              DSM_RS_ABORT_CRC_FAILED
#define DSM_RC_ABORT_INVALID_GROUP_ACTION    DSM_RS_ABORT_INVALID_GROUP_ACTION
#define DSM_RC_ABORT_DISK_UNDEFINED          DSM_RS_ABORT_DISK_UNDEFINED
#define DSM_RC_ABORT_BAD_DESTINATION         DSM_RS_ABORT_BAD_DESTINATION
#define DSM_RC_ABORT_DATAMOVER_NOT_AVAILABLE DSM_RS_ABORT_DATAMOVER_NOT_AVAILABLE
#define DSM_RC_ABORT_STGPOOL_COPY_CONT_NO    DSM_RS_ABORT_STGPOOL_COPY_CONT_NO
#define DSM_RC_ABORT_RETRY_SINGLE_TXN        DSM_RS_ABORT_RETRY_SINGLE_TXN
#define DSM_RC_ABORT_TOC_CREATION_FAIL       DSM_RS_ABORT_TOC_CREATION_FAIL
#define DSM_RC_ABORT_TOC_LOAD_FAIL           DSM_RS_ABORT_TOC_LOAD_FAIL
#define DSM_RC_ABORT_PATH_RESTRICTED         DSM_RS_ABORT_PATH_RESTRICTED
#define DSM_RC_ABORT_NO_LANFREE_SCRATCH      DSM_RS_ABORT_NO_LANFREE_SCRATCH
#define DSM_RC_ABORT_INSERT_NOT_ALLOWED      DSM_RS_ABORT_INSERT_NOT_ALLOWED
#define DSM_RC_ABORT_DELETE_NOT_ALLOWED      DSM_RS_ABORT_DELETE_NOT_ALLOWED
#define DSM_RC_ABORT_TXN_LIMIT_EXCEEDED      DSM_RS_ABORT_TXN_LIMIT_EXCEEDED
#define DSM_RC_ABORT_OBJECT_ALREADY_HELD     DSM_RS_ABORT_OBJECT_ALREADY_HELD
#define DSM_RC_ABORT_INVALID_CHUNK_REFERENCE DSM_RS_ABORT_INVALID_CHUNK_REFERENCE
#define DSM_RC_ABORT_DESTINATION_NOT_DEDUP   DSM_RS_ABORT_DESTINATION_NOT_DEDUP
#define DSM_RC_ABORT_DESTINATION_POOL_CHANGED DSM_RS_ABORT_DESTINATION_POOL_CHANGED
#define DSM_RC_ABORT_NOT_ROOT                DSM_RS_ABORT_NOT_ROOT


/* Definições para códigos de rejeição de conexão ao servidor          */
/* Esses códigos de erro estão no intervalo inclusivo (51 a 99).           */
#define DSM_RC_REJECT_NO_RESOURCES           51
#define DSM_RC_REJECT_VERIFIER_EXPIRED       52
#define DSM_RC_REJECT_ID_UNKNOWN             53
#define DSM_RC_REJECT_DUPLICATE_ID           54
#define DSM_RC_REJECT_SERVER_DISABLED        55
#define DSM_RC_REJECT_CLOSED_REGISTER        56
#define DSM_RC_REJECT_CLIENT_DOWNLEVEL       57
#define DSM_RC_REJECT_SERVER_DOWNLEVEL       58
#define DSM_RC_REJECT_ID_IN_USE              59
#define DSM_RC_REJECT_ID_LOCKED              61
#define DSM_RC_SIGNONREJECT_LICENSE_MAX      62
#define DSM_RC_REJECT_NO_MEMORY              63
#define DSM_RC_REJECT_NO_DB_SPACE            64
#define DSM_RC_REJECT_NO_LOG_SPACE           65
#define DSM_RC_REJECT_INTERNAL_ERROR         66
#define DSM_RC_SIGNONREJECT_INVALID_CLI      67 /* tipo de cliente não licenciado */
#define DSM_RC_CLIENT_NOT_ARCHRETPROT        68
#define DSM_RC_REJECT_LASTSESS_CANCELED      69
```

```
#define DSM_RC_REJECT_UNICODE_NOT_ALLOWED    70
#define DSM_RC_REJECT_NOT_AUTHORIZED         71
#define DSM_RC_REJECT_TOKEN_TIMEOUT          72
#define DSM_RC_REJECT_INVALID_NODE_TYPE      73
#define DSM_RC_REJECT_INVALID_SESSIONINIT    74
#define DSM_RC_REJECT_WRONG_PORT             75
#define DSM_RC_CLIENT_NOT_SPMRETPROT         79



#define DSM_RC_USER_ABORT          101 /* processamento interrompido pelo usuário */
#define DSM_RC_NO_MEMORY           102 /* nenhum RAM restante para concluir o pedido  */
#define DSM_RC_TA_COMM_DOWN       2021 /* não é mais utilizado              */
#define DSM_RC_FILE_NOT_FOUND      104 /* arquivo especificado não localizado */
#define DSM_RC_PATH_NOT_FOUND      105 /* caminho especificado não existe   */
#define DSM_RC_ACCESS_DENIED       106 /* negado devido a permissão imprópria */
#define DSM_RC_NO_HANDLES          107 /* nenhum identificador de arquivo adicional disponível  */
#define DSM_RC_FILE_EXISTS         108 /* arquivo já existente              */
#define DSM_RC_INVALID_PARM        109 /* parâmetro inválido transmitido. CRITICAL*/
#define DSM_RC_INVALID_HANDLE      110 /* identificador de arquivos inválido transmitido */
#define DSM_RC_DISK_FULL           111 /* falat de espaço em disco         */
#define DSM_RC_PROTOCOL_VIOLATION  113 /* violação do protocolo de chamada. CRITICAL    */
#define DSM_RC_UNKNOWN_ERROR       114 /* erro desconhecido do sistema. CRITICAL        */
#define DSM_RC_UNEXPECTED_ERROR    115 /* erro inesperado. CRITICAL    */
#define DSM_RC_FILE_BEING_EXECUTED 116 /* Nenhuma gravação permitida        */
#define DSM_RC_DIR_NO_SPACE        117 /* diretório não pode ser expandido  */
#define DSM_RC_LOOPED_SYM_LINK     118 /* número excessivo de links
                                          encontrados no caminho de conversão.  */
#define DSM_RC_FILE_NAME_TOO_LONG  119 /* nome do arquivo muito longo       */
#define DSM_RC_FILE_SPACE_LOCKED   120 /* espaço no arquivo bloqueado pelo sistema */
#define DSM_RC_FINISHED            121 /* processamento concluído           */
#define DSM_RC_UNKNOWN_FORMAT      122 /* formato desconhecido             */
#define DSM_RC_NO_AUTHORIZATION    123 /* resposta do servidor quando o cliente
                                          não tem autorização para ler os
                                          dados de backup/archive do proprietário de outro host  */
#define DSM_RC_FILE_SPACE_NOT_FOUND 124/* espaço no arquivo especificado não localizado   */
#define DSM_RC_TXN_ABORTED         125 /* transação interrompida           */
#define DSM_RC_SUBDIR_AS_FILE      126 /* Nome do subdiretório existe como arquivo  */
#define DSM_RC_PROCESS_NO_SPACE    127 /* processo não tem mais espaço em disco.   */
#define DSM_RC_PATH_TOO_LONG       128 /* um caminho de diretório que estava sendo construído ficou
                                          muito longo                       */
#define DSM_RC_NOT_COMPRESSED      129 /* arquivo supostamente compactado
                                          não está realmente                */
#define DSM_RC_TOO_MANY_BITS       130 /* arquivo compactado utilizando mais bits
                                          do que o expansor pode manipular  */
#define DSM_RC_SYSTEM_ERROR        131 /* erro interno do sistema           */
#define DSM_RC_NO_SERVER_RESOURCES 132 /* servidor sem recursos.          */
#define DSM_RC_FS_NOT_KNOWN        133 /* o espaço no arquivo não é conhecido
                                          do servidor                       */
#define DSM_RC_NO_LEADING_DIRSEP   134 /* nenhum separador de diretório à esquerda  */
#define DSM_RC_WILDCARD_DIR        135 /* caractere curinga no caminho de
                                          diretório quando não permitido    */
#define DSM_RC_COMM_PROTOCOL_ERROR 136 /* erro do protocolo de comunicações */
#define DSM_RC_AUTH_FAILURE        137 /* falha de autenticação             */
#define DSM_RC_TA_NOT_VALID        138 /* TA não é root e/ou programa SUID  */
#define DSM_RC_KILLED              139 /* processo interrompido.                  */

#define DSM_RC_RETRY               143 /* tente a mesma operação novamente  */

#define DSM_RC_WOULD_BLOCK         145 /* a operação faria com que o sistema
                                          bloqueasse a espera pela entrada.          */
#define DSM_RC_TOO_SMALL           146 /* área pequena para padrão compilado*/
#define DSM_RC_UNCLOSED            147 /* sem colchete direito no padrão    */
#define DSM_RC_NO_STARTING_DELIMITER 148 /* padrão precisa iniciar com
                                            delimitador de diretório        */
#define DSM_RC_NEEDED_DIR_DELIMITER 149 /* um delimitador de diretório é necessário
                                          imediatamente antes e após a
                                          metacadeia "diretórios correspondentes"
                                          ("...") e um não foi encontrado      */
#define DSM_RC_UNKNOWN_FILE_DATA_TYPE 150 /* tipo de dados de arquivo estruturado é
                                             desconhecido                   */
#define DSM_RC_BUFFER_OVERFLOW     151 /* estouro de buffer de dados        */


#define DSM_RC_NO_COMPRESS_MEMORY  154 /* Falta de memória para compactação/expansão */
#define DSM_RC_COMPRESS_GREW       155 /* Compactação cresceu               */
#define DSM_RC_INV_COMM_METHOD     156 /* Método de comunicação especificado inválido */
#define DSM_RC_WILL_ABORT          157 /* Transação será interrompida       */
#define DSM_RC_FS_WRITE_LOCKED     158 /* Espaço no arquivo bloqueado para gravação*/
```

```
#define DSM_RC_SKIPPED_BY_USER     159 /* Arquivo desejado pelo usuário ignorado no
                                       caso de ABORT_DATA_OFFLINE        */
#define DSM_RC_TA_NOT_FOUND        160 /* TA não localizado em seu diretório*/
#define DSM_RC_TA_ACCESS_DENIED    161 /* Acesso a TA negado                */
#define DSM_RC_FS_NOT_READY        162 /* Espaço no arquivo não está pronto */
#define DSM_RC_FS_IS_BAD           163 /* Espaço no arquivo inválido        */
#define DSM_RC_FIO_ERROR           164 /* Erro de entrada/saída do arquivo  */
#define DSM_RC_WRITE_FAILURE       165 /* Erro ao gravar no arquivo         */
#define DSM_RC_OVER_FILE_SIZE_LIMIT 166 /* Arquivo acima do limite do sistema/usuário */
#define DSM_RC_CANNOT_MAKE         167 /* Não foi possível criar arquivo/diretório,
                                       pode ser um nome inválido         */
#define DSM_RC_NO_PASS_FILE        168 /* arquivo de senha necessário e usuário
                                       não é root                        */
#define DSM_RC_VERFILE_OLD         169 /* senha armazenada localmente não
                                       corresponde à do host         */
#define DSM_RC_INPUT_ERROR         173 /* não foi possível ler a entrada do teclado*/
#define DSM_RC_REJECT_PLATFORM_MISMATCH 174 /* O nome da plataforma não corresponde
                                          à plataforma indicada pelo servidor
                                          para o cliente */
#define DSM_RC_TL_NOT_FILE_OWNER   175 /* O usuário que está tentando fazer backup de um arquivo não é
                                       o proprietário do arquivo.                */
#define DSM_RC_COMPRESSED_DATA_CORRUPTED 176 /* Os dados compactados estão corrompidos*/
#define DSM_RC_UNMATCHED_QUOTE     177 /* faltando aspas iniciais ou finais */

#define DSM_RC_SIGNON_FAILOVER_MODE 178      /* Executado failover no servidor de replicação,
                                          em execução no modo de failover */
#define DSM_RC_FAILOVER_MODE_FUNC_BLOCKED 179 /* a função foi bloqueada porque
                                             a sessão está no modo de failover */

/*----------------------------------------------------------------------------*/
/* Códigos de retorno 180-199 são reservados para manipulação do Conjunto de Políticas */
/*----------------------------------------------------------------------------*/
#define DSM_RC_PS_MULTBCG          181 /* Vários grupos de cópias de backup em 1MC*/
#define DSM_RC_PS_MULTACG          182 /* Vários grupos de cópias de archive  em 1 MC*/
#define DSM_RC_PS_NODFLTMC         183 /* Nome de MC padrão não está no conjunto de políticas  */
#define DSM_RC_TL_NOBCG            184 /* Backup req, nenhum grupo de cópias de backup */
#define DSM_RC_TL_EXCLUDED         185 /* Backup req, excl. pelo filtro in/ex */
#define DSM_RC_TL_NOACG            186 /* Archive req, nenhum grupo de cópias de archive */
#define DSM_RC_PS_INVALID_ARCHMC   187 /* Nome de MC inválido na substituição de archive*/
#define DSM_RC_NO_PS_DATA          188 /* Nenhum dado de conjunto de políticas no servidor */
#define DSM_RC_PS_INVALID_DIRMC    189 /* MC de diretório inválida especificada no
                                       arquivo de opções.                 */
#define DSM_RC_PS_NO_CG_IN_DIR_MC  190 /* Nenhum grupo de cópias de backup na MC de diretório.
                                       Uma MC deve ser especificada utilizando-se
                                       .                             */

#define DSM_RC_WIN32_UNSUPPORTED_FILE_TYPE 280 /* Arquivo não é do
                                             tipo Win32 FILE_TYPE_DISK  */

/*----------------------------------------------------------------------------*/
/* Códigos de retorno para o Agente de Comunicação Confiável                  */
/*----------------------------------------------------------------------------*/
#define DSM_RC_TCA_NOT_ROOT        161 /* Acesso a TA negado                 */
#define DSM_RC_TCA_ATTACH_SHR_MEM_ERR  200 /* Erro ao conectar memória compartilhada  */
#define DSM_RC_TCA_SHR_MEM_BLOCK_ERR   200 /* Erro de bloco de memória compartilhada */
#define DSM_RC_TCA_SHR_MEM_IN_USE      200 /* Erro de bloco de memória compartilhada */
#define DSM_RC_TCA_SHARED_MEMORY_ERROR 200 /* Erro de bloco de memória compartilhada */
#define DSM_RC_TCA_SEGMENT_MISMATCH    200 /* Erro de bloco de memória compartilhada */
#define DSM_RC_TCA_FORK_FAILED     292 /* Erro ao bifurcar processo TCA      */
#define DSM_RC_TCA_DIED            294 /* TCA interrompido inesperadamente   */
#define DSM_RC_TCA_INVALID_REQUEST 295 /* Pedido inválido enviado ao TCA     */
#define DSM_RC_TCA_SEMGET_ERROR    297 /* Erro ao obter semáforos            */
#define DSM_RC_TCA_SEM_OP_ERROR    298 /* Erro em configuração ou espera de semáforo */
#define DSM_RC_TCA_NOT_ALLOWED     299 /* TCA não permitido (multiencadeamento) */

/*----------------------------------------------------------------------------*/
/* 400-430 para opções                                                        */
/*----------------------------------------------------------------------------*/
#define DSM_RC_INVALID_OPT           400 /* opção inválida                   */
#define DSM_RC_NO_HOST_ADDR          405 /* Informações insuficientes para conectar servidor  */
#define DSM_RC_NO_OPT_FILE           406 /* Nenhum arquivo de configuração de usuário padrão */
#define DSM_RC_MACHINE_SAME          408 /* -MACHINENAME igual ao nome real  */
#define DSM_RC_INVALID_SERVER        409 /* Nome de servidor inválido do cliente  */
#define DSM_RC_INVALID_KEYWORD       410 /* Palavra-chave de opção inválida  */
#define DSM_RC_PATTERN_TOO_COMPLEX   411 /* Não é possível corresponder entrada de Inclusão/Exclusão*/
#define DSM_RC_NO_CLOSING_BRACKET    412 /* Faltando colchete direito de inc/excl */
#define DSM_RC_OPT_CLIENT_NOT_ACCEPTING 417/* Cliente não aceita esta opção
                                        do servidor              */
#define DSM_RC_OPT_CLIENT_DOES_NOT_WANT 418/* Cliente não deseja este valor
```

```
                                        do servidor              */
#define DSM_RC_OPT_NO_INCLEXCL_FILE  419   /* arquivo inclexcl não localizado*/
#define DSM_RC_OPT_OPEN_FAILURE      420   /* não é possível abrir o arquivo */
#define DSM_RC_OPT_INV_NODENAME      421/* utilizado para o Windows se nodename=local
                                         machine quando CLUSTERNODE=YES       */
#define DSM_RC_OPT_NODENAME_INVALID  423/* nome de nó genérico inválido       */
#define DSM_RC_OPT_ERRORLOG_CONFLICT 424/* logmax e retention especificados */
#define DSM_RC_OPT_SCHEDLOG_CONFLICT 425/* logmax e retention especificados  */
#define DSM_RC_CANNOT_OPEN_TRACEFILE 426/* não é possível abrir arquivo de rastreio  */
#define DSM_RC_CANNOT_OPEN_LOGFILE   427/* não é possível abrir o arquivo do log de erros         */
#define DSM_RC_OPT_SESSINIT_LF_CONFLICT 428/* sessioninit=server e
              enablelanfree=yes são especificados*/
#define DSM_RC_OPT_OPTION_IGNORE     429/* a opção será ignorada            */
#define DSM_RC_OPT_DEDUP_CONFLICT    430/* não é possível abrir o arquivo do log de erros        */
#define DSM_RC_OPT_HSMLOG_CONFLICT 431/* logmax & retention especificados */


/*-----------------------------------------------------------------------------*/
/* 600 a 610 ´para códigos de etiqueta de volume                               */
/*-----------------------------------------------------------------------------*/
#define DSM_RC_DUP_LABEL           600 /* etiqueta de volume duplicada localizada */
#define DSM_RC_NO_LABEL            601 /* unidade não tem etiqueta           */

/*-----------------------------------------------------------------------------*/
/* Códigos de retorno para processamento de arquivo de mensagem                */
/*-----------------------------------------------------------------------------*/
#define DSM_RC_NLS_CANT_OPEN_TXT   610 /* erro ao tentar abrir o arquivo txt de mensagem  */
#define DSM_RC_NLS_CANT_READ_HDR   611 /* erro ao tentar ler cabeçalho       */
#define DSM_RC_NLS_INVALID_CNTL_REC 612 /* registro de controle inválido     */
#define DSM_RC_NLS_INVALID_DATE_FMT 613 /* formato de data padrão inválido   */
#define DSM_RC_NLS_INVALID_TIME_FMT 614 /* formato de hora padrão inválido   */
#define DSM_RC_NLS_INVALID_NUM_FMT 615 /* formato de número padrão inválido  */

/*-----------------------------------------------------------------------------*/
/* Códigos de retorno 620-630 são reservados para os códigos de retorno de mensagem de log */
/*-----------------------------------------------------------------------------*/
#define DSM_RC_LOG_CANT_BE_OPENED  620 /* erro ao tentar abrir log de erros  */
#define DSM_RC_LOG_ERROR_WRITING_TO_LOG 621 /* ocorreu um erro ao gravar no
                                       arquivo de log                */
#define DSM_RC_LOG_NOT_SPECIFIED   622 /* nenhum arquivo de log de erros especificado */


/*-----------------------------------------------------------------------------*/
/* Códigos de retorno 900-999 SOMENTE PARA CLIENTE TSM                   */
/*-----------------------------------------------------------------------------*/
#define DSM_RC_NOT_ADSM_AUTHORIZED  927 /* Deve ser ADSM autorizado para executar*/
                                    /* ação: usuário root ou autorizado por senha  */
#define DSM_RC_REJECT_USERID_UNKNOWN 940 /* ID do usuário desconhecido no servidor     */
#define DSM_RC_FILE_IS_SYMLINK      959 /* log de erros ou rastreio é um link simbólico
                                                                                                                                                */

#define DSM_RC_DIRECT_STORAGE_AGENT_UNSUPPORTED 961 /* Conexão direta a SA não suportada */
#define DSM_RC_FS_NAMESPACE_DOWNLEVEL 963 /* Espaço de nomes longo removido
                                     do volume do Netware */
#define DSM_RC_CONTINUE_NEW_CONSUMER   972 /* Continue o processamento usando um novo consumidor */
#define DSM_RC_CONTINUE_NEW_CONSUMER_NODEDUP   973 /* Continue o processamento usando um novo consumidor sem dedup*/
#define DSM_RC_CONTINUE_NEW_CONSUMER_NOCOMPRESS 976 /* Continue processing using a new consumer no compression */


#define DSM_RC_SERVER_SUPPORTS_FUNC    994 /* o servidor suporta esta função  */
#define DSM_RC_SERVER_AND_SA_SUPPORT_FUNC 995 /* Servidor e SA suportam função */
#define DSM_RC_SERVER_DOWNLEVEL_FUNC   996 /* O servidor tem nível inferior para função  */
#define DSM_RC_STORAGEAGENT_DOWNLEVEL  997 /* o agente de armazenamento tem nível inferior */
#define DSM_RC_SERVER_AND_SA_DOWNLEVEL 998 /* nível inferior de servidor e SA    */


/* Códigos de erro de TCP/IP */
#define DSM_RC_TCPIP_FAILURE       -50 /* Falha de comunicações de TCP/IP       */
#define DSM_RC_CONN_TIMEDOUT       -51 /* Tempo limite de tentativa de conexão de TCP/IP excedido */
#define DSM_RC_CONN_REFUSED        -52 /* Conexão TCP/IP recusada pelo host  */
#define DSM_RC_BAD_HOST_NAME       -53 /* Nome de host TCP/IP inválido especificado */
#define DSM_RC_NETWORK_UNREACHABLE -54 /* Nome do host TCP/IP inacessível    */
#define DSM_RC_WINSOCK_MISSING     -55 /* WINSOCK.DLL de TCP/IP faltando     */
#define DSM_RC_TCPIP_DLL_LOADFAILURE -56 /* Erro de LoadLibrary              */
#define DSM_RC_TCPIP_LOADFAILURE   -57 /* Erro de GetProcAddress   */
#define DSM_RC_TCPIP_USER_ABORT    -58 /* Usuário interrompeu na camada TCP/IP */

/*-----------------------------------------------------------------------------*/
/* Códigos de retorno (-71)-(-90) reservados para códigos de erro CommTSM    */
```

```
/*---------------------------------------------------------------------------*/
#define DSM_RC_TSM_FAILURE         -71 /* Falha de comunicações de TSM       */
#define DSM_RC_TSM_ABORT           -72 /* Sessão interrompida de forma anormal*/

/*Códigos de erro de comm3270 - não mais utilizados*/
#define DSM_RC_COMM_TIMEOUT       2021   /* não é mais utilizado  */
#define DSM_RC_EMULATOR_INACTIVE  2021   /* não é mais utilizado  */
#define DSM_RC_BAD_HOST_ID        2021   /* não é mais utilizado  */
#define DSM_RC_HOST_SESS_BUSY     2021   /* não é mais utilizado  */
#define DSM_RC_3270_CONNECT_FAILURE 2021 /* não é mais utilizado  */
#define DSM_RC_NO_ACS3ELKE_DLL    2021   /* não é mais utilizado  */
#define DSM_RC_EMULATOR_ERROR     2021   /* não é mais utilizado  */
#define DSM_RC_EMULATOR_BACKLEVEL 2021   /* não é mais utilizado  */
#define DSM_RC_CKSUM_FAILURE      2021   /* não é mais utilizado  */


/* Os códigos de retorno a seguir são para EHLLAPI para Windows             */
#define DSM_RC_3270COMMError_DLL        2021   /* não é mais utilizado  */
#define DSM_RC_3270COMMError_GetProc    2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_DLL         2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_GetProc     2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_HostConnect 2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_AllocBuff   2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_SendKey     2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_PacketChk   2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_ChkSum      2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_HostTimeOut 2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_Send        2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_Recv        2021   /* não é mais utilizado  */
#define DSM_RC_EHLLAPIError_General     2021   /* não é mais utilizado  */
#define DSM_RC_PC3270_MISSING_DLL       2021   /* não é mais utilizado  */
#define DSM_RC_3270COMM_MISSING_DLL     2021   /* não é mais utilizado  */


/* Códigos de erro de NETBIOS */
#define DSM_RC_NETB_ERROR         -151 /* Não possível incluir o nó na LAN   */
#define DSM_RC_NETB_NO_DLL        -152 /* O ACSNETB.DLL não pôde ser carregado*/
#define DSM_RC_NETB_LAN_ERR       -155 /* Erro de LAN detectado              */
#define DSM_RC_NETB_NAME_ERR      -158 /* Erro de Netbios em Incluir Nome    */
#define DSM_RC_NETB_TIMEOUT       -159 /* Tempo limite de envio de Netbios   */
#define DSM_RC_NETB_NOTINST       -160 /* Netbios não instalado - DOS        */
#define DSM_RC_NETB_REBOOT        -161 /* Erro de configuração de Netbios - reinicializar DOS    */

/* Códigos de erro de Canal Denominado */
#define DSM_RC_NP_ERROR                       -190

/* Códigos de erro de CPIC */
#define DSM_RC_CPIC_ALLOCATE_FAILURE          2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_TYPE_MISMATCH             2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PIP_NOT_SPECIFY_ERR       2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_SECURITY_NOT_VALID        2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_SYNC_LVL_NO_SUPPORT       2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_TPN_NOT_RECOGNIZED        2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_TP_ERROR                  2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PARAMETER_ERROR           2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PROD_SPECIFIC_ERR         2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PROGRAM_ERROR             2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_RESOURCE_ERROR            2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_DEALLOCATE_ERROR          2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_SVC_ERROR                 2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PROGRAM_STATE_CHECK       2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_PROGRAM_PARAM_CHECK       2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_UNSUCCESSFUL              2021 /* não é mais utilizado  */
#define DSM_RC_UNKNOWN_CPIC_PROBLEM           2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_MISSING_LU                2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_MISSING_TP                2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_SNA6000_LOAD_FAIL         2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_STARTUP_FAILURE           2021 /* não é mais utilizado  */

/*---------------------------------------------------------------------------*/
/* Códigos de retorno -300 a -307 são reservados para comunicações IPX/SPX   */
/*---------------------------------------------------------------------------*/
#define DSM_RC_TLI_ERROR                      2021 /* não é mais utilizado  */
#define DSM_RC_IPXSPX_FAILURE                 2021 /* não é mais utilizado  */
#define DSM_RC_TLI_DLL_MISSING                2021 /* não é mais utilizado  */
#define DSM_RC_DLL_LOADFAILURE                2021 /* não é mais utilizado  */
#define DSM_RC_DLL_FUNCTION_LOADFAILURE       2021 /* não é mais utilizado  */
#define DSM_RC_IPXCONN_REFUSED                2021 /* não é mais utilizado  */
#define DSM_RC_IPXCONN_TIMEDOUT               2021 /* não é mais utilizado  */
```

```
#define DSM_RC_IPXADDR_UNREACHABLE            2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_MISSING_DLL               2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_DLL_LOADFAILURE           2021 /* não é mais utilizado  */
#define DSM_RC_CPIC_FUNC_LOADFAILURE          2021 /* não é mais utilizado  */

/*=== Códigos de erro do Protocolo de Memória Compartilhada ===*/
#define DSM_RC_SHM_TCPIP_FAILURE              -450
#define DSM_RC_SHM_FAILURE                    -451
#define DSM_RC_SHM_NOTAUTH                    -452

#define DSM_RC_NULL_OBJNAME       2000 /* Ponteiro do nome do objeto é NULL*/
#define DSM_RC_NULL_DATABLKPTR    2001 /* dataBlkPtr é NULL               */
#define DSM_RC_NULL_MSG           2002 /* parâmetro de mensagem em dsmRCMsg é NULL     */

#define DSM_RC_NULL_OBJATTRPTR    2004 /* Ponteiro do Atributo de Objeto é NULL  */

#define DSM_RC_NO_SESS_BLK        2006 /* nenhuma informação de sessão do servidor */
#define DSM_RC_NO_POLICY_BLK      2007 /* nenhuma informação de hdr de política*/
#define DSM_RC_ZERO_BUFLEN        2008 /* bufferLen é zero para dataBlkPtr */
#define DSM_RC_NULL_BUFPTR        2009 /* bufferPtr é NULL para dataBlkPtr */

#define DSM_RC_INVALID_OBJTYPE    2010 /* tipo de objeto inválido          */
#define DSM_RC_INVALID_VOTE       2011 /* voto inválido                    */
#define DSM_RC_INVALID_ACTION     2012 /* ação inválida                    */
#define DSM_RC_INVALID_DS_HANDLE  2014 /* identificador de ADSM inválido   */
#define DSM_RC_INVALID_REPOS      2015 /* valor inválido para o repositório*/
#define DSM_RC_INVALID_FSNAME     2016 /* fs deve iniciar com delimitador de diretório */
#define DSM_RC_INVALID_OBJNAME    2017 /* nome de caminho completo inválido*/
#define DSM_RC_INVALID_LLNAME     2018 /* ll deve iniciar com delimitador de diretório */
#define DSM_RC_INVALID_OBJOWNER   2019 /* nome do proprietário do objeto inválido*/
#define DSM_RC_INVALID_ACTYPE     2020 /* tipo de ação inválido            */
#define DSM_RC_INVALID_RETCODE    2021 /* dsmRC em dsmRCMsg é inválido     */
#define DSM_RC_INVALID_SENDTYPE   2022 /* tipo de envio inválido           */
#define DSM_RC_INVALID_PARAMETER  2023 /* parâmetro inválido               */
#define DSM_RC_INVALID_OBJSTATE   2024 /* ativo, inativo ou correspondência?  */
#define DSM_RC_INVALID_MCNAME     2025 /* Nome da classe de gerenciamento não localizado */
#define DSM_RC_INVALID_DRIVE_CHAR 2026 /* Letra da unidade não é alfabética*/
#define DSM_RC_NULL_FSNAME        2027 /* Nomes do espaço no arquivo é NULL*/
#define DSM_RC_INVALID_HLNAME     2028 /* hl deve iniciar com delimitador de diretório */

#define DSM_RC_NUMOBJ_EXCEED      2029 /* Número de objetos de BeginGetData excedido   */

#define DSM_RC_NEWPW_REQD         2030 /* nova senha requerida             */
#define DSM_RC_OLDPW_REQD         2031 /* senha antiga requerida           */
#define DSM_RC_NO_OWNER_REQD      2032 /* proprietário não permitido. Permitir padrão */
#define DSM_RC_NO_NODE_REQD       2033 /* nó não permitido c/ pw=generate  */
#define DSM_RC_KEY_MISSING        2034 /* arquivo de chave não pode ser localizado */
#define DSM_RC_KEY_BAD            2035 /* conteúdo do arquivo de chave é inválido */

#define DSM_RC_BAD_CALL_SEQUENCE  2041 /* Seqüência de chamadas de DSM não permitida*/
#define DSM_RC_INVALID_TSMBUFFER  2042 /* valor inválido para tsmbuffhandle ou dataPtr */
#define DSM_RC_TOO_MANY_BYTES     2043 /* número excessivo de bytes copiado para o buffer  */
#define DSM_RC_MUST_RELEASE_BUFFER 2044 /* não é possível sair do aplicativo, é necessário liberar */
                                        /* buffers */
#define DSM_RC_BUFF_ARRAY_ERROR   2045 /* erro interno da matriz de buffer */
#define DSM_RC_INVALID_DATABLK    2046 /* utilizando tsmbuff, datablk deveria ser nulo */
#define DSM_RC_ENCR_NOT_ALLOWED   2047 /* ao utilizar tsmbuffers, criptografia não permitida */
#define DSM_RC_OBJ_COMPRESSED     2048 /* Não é possível restaurar utilizando tsmBuff no objeto */
                                       /* compactado */
#define DSM_RC_OBJ_ENCRYPTED      2049 /* Não é possível restaurar um objeto criptografado */
                                       /* utilizando tsmbuff */
#define DSM_RC_WILDCHAR_NOTALLOWED 2050 /* Curinga não permitido para hl,ll  */
#define DSM_RC_POR_NOT_ALLOWED    2051 /* Não é possível utilizar restauração parcial do objeto */
                                       /* com tsmBuffers */
#define DSM_RC_NO_ENCRYPTION_KEY  2052 /* Chave de criptografia não localizada*/
#define DSM_RC_ENCR_CONFLICT      2053 /* opções mutuamente exclusivas */

#define DSM_RC_FSNAME_NOTFOUND    2060 /* Nome do espaço no arquivo não localizado*/
#define DSM_RC_FS_NOT_REGISTERED  2061 /* Nome do espaço no arquivo não registrado */
#define DSM_RC_FS_ALREADY_REGED   2062 /* Espaço no arquivo já registrado  */
#define DSM_RC_OBJID_NOTFOUND     2063 /* Nenhum ID de objeto a restaurar  */
#define DSM_RC_WRONG_VERSION      2064 /* Nível de código errado           */
#define DSM_RC_WRONG_VERSION_PARM 2065 /* Nível errado de estrutura do parâmetro  */

#define DSM_RC_NEEDTO_ENDTXN      2070 /* Necessário chamar dsmEndTxn           */

#define DSM_RC_OBJ_EXCLUDED       2080 /* Objeto excluído por MC         */
#define DSM_RC_OBJ_NOBCG          2081 /* Objeto não possui nenhum grupo de cópias de backup  */
#define DSM_RC_OBJ_NOACG          2082 /* Objeto não possui nenhum grupo de cópias de archive */
```

```
#define DSM_RC_APISYSTEM_ERROR     2090 /* Erro interna da API             */

#define DSM_RC_DESC_TOOLONG             2100 /* descrição muito longa           */
#define DSM_RC_OBJINFO_TOOLONG          2101 /* atributo de objeto objinfo muito longo     */
#define DSM_RC_HL_TOOLONG               2102 /* Qualificador de nível superior muito longo */
#define DSM_RC_PASSWD_TOOLONG           2103 /* senha muito longa              */
#define DSM_RC_FILESPACE_TOOLONG        2104 /* nome do espaço no arquivo é muito longo */
#define DSM_RC_LL_TOOLONG               2105 /* Qualificador de nível inferior é muito longo */
#define DSM_RC_FSINFO_TOOLONG           2106 /* comprimento do espaço no arquivo é muito grande */
#define DSM_RC_SENDDATA_WITH_ZERO_SIZE  2107 /* enviar dados c/ estimativa igual a zero */

/*=== novos códigos de retorno para dsmaccess ===*/
#define DSM_RC_INVALID_ACCESS_TYPE 2110 /* tipo de acesso inválido          */
#define DSM_RC_QUERY_COMM_FAILURE  2111 /* erro de comunicação durante a consulta */
#define DSM_RC_NO_FILES_BACKUP     2112 /* Nenhum arquivo com backup para este fs   */
#define DSM_RC_NO_FILES_ARCHIVE    2113 /* Nenhum arquivo arquivado para este fs    */
#define DSM_RC_INVALID_SETACCESS   2114 /* formato de acesso configurado inválido */

/*=== novos códigos de retorno para dsmaccess ===*/
#define DSM_RC_STRING_TOO_LONG     2120 /* Parâmetro da cadeia muito longo   */

#define DSM_RC_MORE_DATA           2200 /* Há mais dados para restaurar   */

#define DSM_RC_BUFF_TOO_SMALL      2210 /* Buffer de DataBlk muito pequeno para consulta */

#define DSM_RC_NO_API_CONFIGFILE   2228 /*arquivo de configuração da API especificado não localizado*/
#define DSM_RC_NO_INCLEXCL_FILE    2229 /* arquivo inclexcl especificado não localizado*/
#define DSM_RC_NO_SYS_OR_INCLEXCL  2230 /* arquivo dsm.sys ou inclexcl
                                           especificado em dsm.sys não localizado */
#define DSM_RC_REJECT_NO_POR_SUPPORT 2231 /* servidor não tem suporte a POR*/

#define DSM_RC_NEED_ROOT           2300 /* responsável pela chamada da API deve ser root          */
#define DSM_RC_NEEDTO_CALL_BINDMC  2301 /* dsmBindMC deve ser chamado primeiro  */
#define DSM_RC_CHECK_REASON_CODE   2302 /* verifique o código de razão de dsmEndTxn */
#define DSM_RC_NEEDTO_ENDTXN_DEDUP_SIZE_EXCEEDED 2303 /* bytes máximos de dedup excedidos */

/*=== códigos de retorno 2400 - 2410 utilizados pelo arquivo lic, consulte agentrc.h ===*/

/*=== códigos de retorno 2410 - 2430 utilizados pelo agente Oracle, consulte agentrc.h ===*/

#define DSM_RC_ENC_WRONG_KEY       4580 /* a chave fornecida está incorreta */
#define DSM_RC_ENC_NOT_AUTHORIZED  4582 /* usuário não tem permissão para decriptografar  */
#define DSM_RC_ENC_TYPE_UNKNOWN    4584 /* tipo de criptografia desconhecido*/

/*============================================================================
   Códigos de retorno (4600)-(4624) reservados para armazenamento em cluster
============================================================================*/
#define DSM_RC_CLUSTER_INFO_LIBRARY_NOT_LOADED        4600
#define DSM_RC_CLUSTER_LIBRARY_INVALID                4601
#define DSM_RC_CLUSTER_LIBRARY_NOT_LOADED             4602
#define DSM_RC_CLUSTER_NOT_MEMBER_OF_CLUSTER          4603
#define DSM_RC_CLUSTER_NOT_ENABLED                    4604
#define DSM_RC_CLUSTER_NOT_SUPPORTED                  4605
#define DSM_RC_CLUSTER_UNKNOWN_ERROR                  4606


/*============================================================================
   Códigos de retorno (5701)-(5749) reservados para proxy
============================================================================*/
#define DSM_RC_PROXY_REJECT_NO_RESOURCES              5702
#define DSM_RC_PROXY_REJECT_DUPLICATE_ID              5705
#define DSM_RC_PROXY_REJECT_ID_IN_USE                 5710
#define DSM_RC_PROXY_REJECT_INTERNAL_ERROR            5717
#define DSM_RC_PROXY_REJECT_NOT_AUTHORIZED            5722
#define DSM_RC_PROXY_INVALID_FROMNODE                 5746
#define DSM_RC_PROXY_INVALID_SERVERFREE               5747
#define DSM_RC_PROXY_INVALID_CLUSTER                  5748
#define DSM_RC_PROXY_INVALID_FUNCTION                 5749

/*============================================================================
   Códigos de retorno 5801 - 5849 são reservados para criptografia/segurança
============================================================================*/

#define DSM_RC_CRYPTO_ICC_ERROR                       5801
#define DSM_RC_CRYPTO_ICC_CANNOT_LOAD                 5802
#define DSM_RC_SSL_NOT_SUPPORTED                      5803
#define DSM_RC_SSL_INIT_FAILED                        5804
#define DSM_RC_SSL_KEYFILE_OPEN_FAILED                5805
```

```
#define DSM_RC_SSL_KEYFILE_BAD_PASSWORD              5806
#define DSM_RC_SSL_BAD_CERTIFICATE                   5807

/*============================================================================
   Códigos de retorno 6300 - 6399 são reservados para deduplicação do lado do cliente
============================================================================*/
#define DSM_RC_DIGEST_VALIDATION_ERROR               6300  /* Erro de validação de compilação de ponta a ponta  */
#define DSM_RC_DATA_FINGERPRINT_ERROR                6301 /* Falha na impressão digital de Rabin    */
#define DSM_RC_DATA_DEDUP_ERROR                      6302 /* Erro ao converter dados em partes */


#endif /* _H_DSMRC */
```

Referências relacionadas:

Códigos de Retorno da API

Apêndice B. Arquivos de Origem de Definições de Tipos de API

Este apêndice contém definições de estrutura, definições de tipo e constantes para a API. Os primeiros arquivos de cabeçalho, `dsmapitd.h` e `tsmapitd.h`, ilustram as definições que são comuns a todos os sistemas operacionais.

O segundo arquivo de cabeçalho, `dsmapips.h`, fornece um exemplo de definições que são específicas de um determinado sistema operacional, neste exemplo, a plataforma Windows.

O terceiro arquivo de cabeçalho, `release.h`, inclui as informações de versão e de liberação.

As informações fornecidas aqui contêm uma cópia point-in-time dos arquivos que são distribuídos com a API. Visualize os arquivos no pacote de distribuição da API para a versão mais recente.

```
/*****************************************************************
* Tivoli Storage Manager                                         *
* Componente do Cliente da API                                   *
*                                                                *
* (C) Copyright IBM Corporation 1993,2010                        *
*****************************************************************/

/*****************************************************************************
* Nome do Arquivo de Cabeçalho: dsmapitd.h
*
* Ambiente:        ************************************************
*                  ** Este é um arquivo de origem independente   **
*                  ** de plataforma                              **
*
*                  ************************************************
*
* Notas de Design: Este arquivo contém tipos e constantes de dados básicos
*                  que podem ser incluídos por todos os arquivos de origem do cliente. As constantes
do arquivo devem ser
*                  configuradas corretamente para a
máquina ou o sistema
*                  operacional específico no qual o
software cliente
*                  deve ser executado.
*
*                  Definições específicas da plataforma estão incluídas em
*                  dsmapips.h
*
* Nome descritivo: Definições para constantes de API do Tivoli Storage
* Manager
*----------------------------------------------------------------------------*/

#ifndef _H_DSMAPITD
#define _H_DSMAPITD


#include "dsmapips.h"      /* Platform specific definitions*/
#include "release.h"

/*=== configure o alinhamento da estrutura para compactar as estruturas ===*/
#if (_OPSYS_TYPE == DS_WINNT) && !defined(_WIN64)
#pragma pack(1)
#endif
```

```
#ifdef _MAC
/*=============================================================================
  As opções são:
http://developer.apple.com/documentation/DeveloperTools/Conceptual/PowerPCRuntime/Data/chapter_2_section_3.html

#pragma option align=<mode>
em que <mode> é potência, mac68k, natural ou compactado.
=============================================================================*/
#pragma options align = packed
#endif

typedef char osChar_t;

/*<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>*/
/*                     D E F I N I R                                        */
/*<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>*/
/*-------------------------------------------------------------------------+
|  Versão, Release e Nível da API a utilizar em dsmApiVersion em dsmInit() |
+-------------------------------------------------------------------------*/
#define DSM_API_VERSION     COMMON_VERSION
#define DSM_API_RELEASE     COMMON_RELEASE
#define DSM_API_LEVEL       COMMON_LEVEL
#define DSM_API_SUBLEVEL    COMMON_SUBLEVEL

/*-------------------------------------------------------------------------+
| Comprimentos máximos dos campos                                          |
+-------------------------------------------------------------------------*/
#define DSM_MAX_CG_DEST_LENGTH        30    /* destino do grupo de cópias */
#define DSM_MAX_CG_NAME_LENGTH        30    /* nome do grupo de cópias    */
#define DSM_MAX_DESCR_LENGTH          255   /* descrição do archive       */
#define DSM_MAX_DOMAIN_LENGTH         30    /* nome do domínio de política*/
#define DSM_MAX_FSINFO_LENGTH         500   /* inf. do espaço no arquivo  */
#define DSM_MAX_USER_FSINFO_LENGTH    480   /* inf. do máx. de espaço no arquivo do usuário */
#define DSM_MAX_FSNAME_LENGTH         1024  /* nome do espaço no arquivo  */
#define DSM_MAX_FSTYPE_LENGTH         32    /* tipo do espaço no arquivo  */
#define DSM_MAX_HL_LENGTH             1024  /* nome do nível superior do objeto */
#define DSM_MAX_ID_LENGTH             64    /* nome do nó da sessão       */
#define DSM_MAX_LL_LENGTH             256   /* nome do nível inferior do objeto */
#define DSM_MAX_MC_NAME_LENGTH        30    /* nome da classe de ger.     */
#define DSM_MAX_OBJINFO_LENGTH        255   /* informações do objeto      */
#define DSM_MAX_EXT_OBJINFO_LENGTH    1500      /* Extended object info   */
#define DSM_MAX_OWNER_LENGTH          64    /* nome do proprietário do obj*/
#define DSM_MAX_PLATFORM_LENGTH       16    /* tipo do aplicativo         */
#define DSM_MAX_PS_NAME_LENGTH        30    /* nome do conj. de políticas */
#define DSM_MAX_SERVERTYPE_LENGTH     32    /* tipo de plat. do servidor  */
#define DSM_MAX_VERIFIER_LENGTH       64    /* senha                      */
#define DSM_PATH_MAX                  1024  /* cam. do arq de conf da API */
#define DSM_NAME_MAX                  255   /* nome do arq de conf da API */
#define DSM_MAX_NODE_LENGTH           64    /* nome do nó/máquina         */
#define DSM_MAX_RC_MSG_LENGTH         1024  /* par. de mens. para dsmRCMsg*/
#define DSM_MAX_SERVER_ADDRESS        1024  /* endereço do servidor       */

#define DSM_MAX_MC_DESCR_LENGTH       DSM_MAX_DESCR_LENGTH /* classe de ger.*/
#define DSM_MAX_SERVERNAME_LENGTH     DSM_MAX_ID_LENGTH /* nome do servidor */
#define DSM_MAX_GET_OBJ               4080     /* máx. de obj. em BeginGetData */
#define DSM_MAX_PARTIAL_GET_OBJ       1300 /* máx. de obj. parciais em BeginGetData */
#define DSM_MAX_COMPRESSTYPE_LENGTH   32       /* max compression algorithm name */

/*-------------------------------------------------------------------------+
| Comprimentos mínimos dos campos                                          |
+-------------------------------------------------------------------------*/
#define DSM_MIN_COMPRESS_SIZE  2048 /* número mínimo de bytes que um objeto*/
                                    /* precisa antes de permitir a compactação*/

/*-------------------------------------------------------------------------+
|  Valores para mtFlag na chamada de dsmSetup                              |
+-------------------------------------------------------------------------*/
#define DSM_MULTITHREAD      bTrue
#define DSM_SINGLETHREAD     bFalse
```

```
/*--------------------------------------------------------------------------+
|  Valores para o tipo de objeto na estrutura dsmObjName                    |
|  Nota: Esses valores devem ser mantidos em sincronização com dsmcomm.h    |
+--------------------------------------------------------------------------*/
#define DSM_OBJ_FILE                0x01 /*o objeto tem informações & dados de atributos*/
#define DSM_OBJ_DIRECTORY           0x02 /*obj. tem somente inf. do atrib.*/
#define DSM_OBJ_RESERVED1           0x04 /* para utilização futura        */
#define DSM_OBJ_RESERVED2           0x05 /* para utilização futura        */
#define DSM_OBJ_RESERVED3           0x06 /* para utilização futura        */
#define DSM_OBJ_WILDCARD            0xFE /* qualquer tipo de objeto       */
#define DSM_OBJ_ANY_TYPE            0xFF /* para utilização futura        */

/*--------------------------------------------------------------------------+
| Definição de tipo para compressedState em QryResp                         |
+--------------------------------------------------------------------------*/
#define DSM_OBJ_COMPRESSED_UNKNOWN    0
#define DSM_OBJ_COMPRESSED_YES        1
#define DSM_OBJ_COMPRESSED_NO         2

/*--------------------------------------------------------------------------+
| Definições para o campo "tipo de grupo" em tsmGrouphandlerIn_t       |
+--------------------------------------------------------------------------*/

#define  DSM_GROUPTYPE_NONE         0x00   /* Não é um membro do grupo      */
#define  DSM_GROUPTYPE_RESERVED1    0x01   /* para utilização futura        */
#define  DSM_GROUPTYPE_PEER         0x02   /* Grupo de mesmo nível          */
#define  DSM_GROUPTYPE_RESERVED2    0x03   /* para utilização futura        */

/*--------------------------------------------------------------------------+
| Definições para o campo "tipo de membro" em tsmGrouphandlerIn_t      |
+--------------------------------------------------------------------------*/

#define  DSM_MEMBERTYPE_LEADER      0x01   /* líder do grupo  */
#define  DSM_MEMBERTYPE_MEMBER      0x02   /* membro do grupo */

/*--------------------------------------------------------------------------+
| Definições para o campo "tipo de operação" em tsmGrouphandlerIn_t    |
+--------------------------------------------------------------------------*/
#define DSM_GROUP_ACTION_BEGIN         0x01
#define DSM_GROUP_ACTION_OPEN          0x02 /* criar um novo grupo            */
#define DSM_GROUP_ACTION_CLOSE         0x03 /* confirmar e salvar um grupo aberto */
#define DSM_GROUP_ACTION_ADD           0x04 /* anexar a um grupo */
#define DSM_GROUP_ACTION_ASSIGNTO      0x05 /* designar a outro grupo */
#define DSM_GROUP_ACTION_REMOVE        0x06 /* remover um membro de um grupo */

/*--------------------------------------------------------------------------+
| Valores para copySer em estruturas DetailCG para resposta da Classe de   |
| Gerenciamento de Consulta                                                 |
+--------------------------------------------------------------------------*/
#define Copy_Serial_Static          1   /*Estática da Serialização de Cópia*/
#define Copy_Serial_Shared_Static   2   /*Estática Compartilhada da Serialização de Cópia*/
#define Copy_Serial_Shared_Dynamic  3   /*Dinâmica Compartilhada de Serialização de Cópia*/
#define Copy_Serial_Dynamic         4   /*Dinâmica de Serialização de Cópia*/

/*--------------------------------------------------------------------------+
| Valores para copyMode em estruturas DetailCG para resposta da Classe de  |
| Gerenciamento de Consulta                                                 |
+--------------------------------------------------------------------------*/
#define Copy_Mode_Modified          1   /*Modo de Cópia Modificado        */
#define Copy_Mode_Absolute          2   /*Modo de Cópia Absoluto          */

/*--------------------------------------------------------------------------+
|  Valores para objState na estrutura qryBackupData                         |
+--------------------------------------------------------------------------*/
#define DSM_ACTIVE          0x01      /* consultar somente objetos ativos */
#define DSM_INACTIVE        0x02     /* consultar somente objetos inativos*/
#define DSM_ANY_MATCH       0xFF  /* consultar todos os objetos de backup */

/*--------------------------------------------------------------------------+
| Valores limites para o campo dsmDate.year na estrutura qryArchiveData    |
```

```
+---------------------------------------------------------------------------*/
#define DATE_MINUS_INFINITE         0x0000      /* limite inferior        */
#define DATE_PLUS_INFINITE          0xFFFF      /* limite superior        */

/*---------------------------------------------------------------------------+
| Máscaras de bits para o parâmetro de ação de atualização em dsmUpdateFS()|
+---------------------------------------------------------------------------*/
#define DSM_FSUPD_FSTYPE              ((unsigned) 0x00000002)
#define DSM_FSUPD_FSINFO              ((unsigned) 0x00000004)
#define DSM_FSUPD_BACKSTARTDATE       ((unsigned) 0x00000008)
#define DSM_FSUPD_BACKCOMPLETEDATE    ((unsigned) 0x00000010)
#define DSM_FSUPD_OCCUPANCY           ((unsigned) 0x00000020)
#define DSM_FSUPD_CAPACITY            ((unsigned) 0x00000040)
#define DSM_FSUPD_RESERVED1           ((unsigned) 0x00000100)

/*---------------------------------------------------------------------------+
| Máscara de bits para o parâmetro de ação de atualização de backup em     |
| dsmUpdateObj()                                                           |
+---------------------------------------------------------------------------*/
#define DSM_BACKUPD_OWNER              ((unsigned) 0x00000001)
#define DSM_BACKUPD_OBJINFO            ((unsigned) 0x00000002)
#define DSM_BACKUPD_MC                 ((unsigned) 0x00000004)

#define DSM_ARCHUPD_OWNER              ((unsigned) 0x00000001)
#define DSM_ARCHUPD_OBJINFO            ((unsigned) 0x00000002)
#define DSM_ARCHUPD_DESCR              ((unsigned) 0x00000004)

/*---------------------------------------------------------------------------+
|  Valores para o parâmetro de repositório em dsmDeleteFS()                |
+---------------------------------------------------------------------------*/
#define DSM_ARCHIVE_REP       0x0A      /* repositório de archive         */
#define DSM_BACKUP_REP        0x0B      /* repositório de backup          */
#define DSM_REPOS_ALL         0x01      /* todos os tipos de repositórios */

/*---------------------------------------------------------------------------+
|  Valores para o parâmetro vote em dsmEndTxn()                             |
+---------------------------------------------------------------------------*/
#define DSM_VOTE_COMMIT  1             /* confirmar transação atual       */
#define DSM_VOTE_ABORT   2             /* recuperar transação atual       */

/*---------------------------------------------------------------------------+
|  Valores para vários sinalizadores retornados na estrutura ApiSessInfo.           |
+---------------------------------------------------------------------------*/
/* Códigos de campos de compactação do cliente */
#define COMPRESS_YES    1        /* cliente deve compactar dados          */
#define COMPRESS_NO     2        /* cliente NÃO deve compactar dados      */
#define COMPRESS_CD     3        /* determinado pelo cliente              */

/* Códigos de permissão de exclusão de archive. */
#define ARCHDEL_YES     1        /* exclusão de archive permitida         */
#define ARCHDEL_NO      2        /* exclusão de archive NÃO permitida     */

/* Códigos de permissão de exclusão de backup. */
#define BACKDEL_YES     1         /* exclusão de backup permitida         */
#define BACKDEL_NO      2         /* exclusão de backup NÃO permitida     */


/*---------------------------------------------------------------------------+
  Valores para vários sinalizadores retornados na estrutura optStruct.           |
----------------------------------------------------------------------------*/
#define DSM_PASSWD_GENERATE  1
#define DSM_PASSWD_PROMPT    0

#define DSM_COMM_TCP        1    /* tcpip         */
#define DSM_COMM_NAMEDPIPE  2    /* Canais nomeados   */
#define DSM_COMM_SHM        3    /* Memória Compartilhada */

/* commmethods obsoletos */
#define DSM_COMM_PVM_IUCV   12
#define DSM_COMM_3270       12
```

```
#define DSM_COMM_IUCV       12
#define DSM_COMM_PWSCS      12
#define DSM_COMM_SNA_LU6_2  12
#define DSM_COMM_IPXSPX     12    /* Para suporte IPX/SPX */
#define DSM_COMM_NETBIOS    12    /* NETBIOS */
#define DSM_COMM_400COMM    12
#define DSM_COMM_CLIO      12    /* CLIO/S */
/*------------------------------------------------------------------------+
|  Valores para userNameAuthorities em dsmInitEx para utilização futura   |
+------------------------------------------------------------------------*/
#define DSM_USERAUTH_NONE     ((dsInt16_t)0x0000)
#define DSM_USERAUTH_ACCESS   ((dsInt16_t)0x0001)
#define DSM_USERAUTH_OWNER    ((dsInt16_t)0x0002)
#define DSM_USERAUTH_POLICY   ((dsInt16_t)0x0004)
#define DSM_USERAUTH_SYSTEM   ((dsInt16_t)0x0008)

/*------------------------------------------------------------------------+
|  Valores para encryptionType em dsmEndSendObjEx, queryResp              |
+------------------------------------------------------------------------*/
#define DSM_ENCRYPT_NO              ((dsUint8_t)0x00)
#define DSM_ENCRYPT_USER            ((dsUint8_t)0x01)
#define DSM_ENCRYPT_CLIENTENCRKEY   ((dsUint8_t)0x02)
#define DSM_ENCRYPT_DES_56BIT       ((dsUint8_t)0x04)
#define DSM_ENCRYPT_AES_128BIT      ((dsUint8_t)0x08)
#define DSM_ENCRYPT_AES_256BIT      ((dsUint8_t)0x10)

/*------------------------------------------------------------------------+
| Definições para o campo mediaClass.                                     |
+------------------------------------------------------------------------*/
/*
 *  As constantes a seguir definem uma hierarquia de classe de acesso
 *  de mídia.
 *  Os números inferiores indicam a mídia que pode fornecer acesso
 *  mais rápido aos dados.
 */

/* Fixa: representa a classe de mídia fixa on-line (como
          discos rígidos). */
#define  MEDIA_FIXED          0x10

/* Biblioteca: representa a classe de mídia montável acessível
            através de um dispositivo de montagem mecânico. */
#define  MEDIA_LIBRARY        0x20

/* utilização futura */
#define  MEDIA_NETWORK        0x30

/* utilização futura */
#define  MEDIA_SHELF          0x40

/* utilização futura */
#define  MEDIA_OFFSITE        0x50

/* utilização futura */
#define  MEDIA_UNAVAILABLE    0xF0


/*------------------------------------------------------------------------+
| Definição de tipo para dados do objeto parcial para dsmBeginGetData()   |
+------------------------------------------------------------------------*/
typedef struct
{
    dsUint16_t   stVersion;                      /* versão da estrutura   */
    dsStruct64_t   partialObjOffset;   /* deslocamento no objeto para iniciar leitura */
    dsStruct64_t   partialObjLength;   /* quantidade do objeto a ler      */
} PartialObjData ;              /* dados do objeto parcial              */

#define PartialObjDataVersion 1  /*                                      */

/*------------------------------------------------------------------------+
```

```
|  Definição de tipo para estrutura de data                              |
+-----------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t    year;            /* ano, inteiro de 16 bits (ex., 1990) */
   dsUint8_t     month;           /* mês, inteiro de 8 bits (1 - 12)     */
   dsUint8_t     day;                  /* dia, inteiro de 8 bits (1 - 31)      */
   dsUint8_t     hour;            /* hora, inteiro de 8 bits (0 - 23)    */
   dsUint8_t     minute;          /* minuto, inteiro de 8 bits (0 - 59)  */
   dsUint8_t     second;          /* segundo, inteiro de 8 bits (0 - 59) */
}dsmDate ;

/*-----------------------------------------------------------------------+
|  Definição de tipo para o ID do objeto em dsmGetObj() e na estrutura   |
|  dsmGetList                                                            |
+-----------------------------------------------------------------------*/
typedef dsStruct64_t  ObjID ;

/*-----------------------------------------------------------------------+
| Definição de tipo para dsmQueryBuff em dsmBeginQuery()                 |
+-----------------------------------------------------------------------*/
typedef void dsmQueryBuff ;

/*-----------------------------------------------------------------------+
| Definição de tipo para o parâmetro dsmGetType em dsmBeginGetData()     |
+-----------------------------------------------------------------------*/
typedef enum
{
        gtBackup = 0x00,                  /* Tipo de processo de backup */
        gtArchive                         /* Tipo de processo de archive*/
} dsmGetType ;

/*-----------------------------------------------------------------------+
|  Definição de tipo para o parâmetro dsmQueryType em dsmBeginQuery()    |
+-----------------------------------------------------------------------*/
typedef enum
{
   qtArchive = 0x00,    /* Tipo de consulta de archive                  */
   qtBackup,            /* Tipo de consulta de backup                   */
   qtBackupActive,      /* Consulta rápida para arquivos de backup ativos */
   qtFilespace,         /* Tipo de consulta de espaço no arquivo        */
   qtMC,                             /* Mgmt. cons de classe de ger */
   qtReserved1,         /* Utilização futura                            */
   qtReserved2,         /* Utilização futura                            */
   qtReserved3,         /* Utilização futura                            */
   qtReserved4,         /* Utilização futura                            */
   qtBackupGroups,      /* Líderes de grupos em um fs específico        */
   qtOpenGroups,        /* Grupos abertos em um fs específico           */
   qtReserved5,         /* Utilização futura                            */
   qtProxyNodeAuth,     /* Nós para os quais o nó pode efetuar proxy    */
   qtProxyNodePeer,                  /* Nós no mesmo nível com o mesmo destino    */
   qtReserved6,                      /* uso futuro                      */
   qtReserved7,                      /* uso futuro                      */
   qtReserved8                       /* uso futuro                      */
}dsmQueryType ;

/*-----------------------------------------------------------------------+
|  Parâmetro sendType da definição de tipo em dsmBindMC() e dsmSendObj()  |
+-----------------------------------------------------------------------*/
typedef enum
{
   stBackup = 0x00,                     /* Tipo de processo de backup  */
   stArchive,                           /* Tipo de processo de archive */
   stBackupMountWait,        /* Processo de backup com mountwait ativado */
   stArchiveMountWait        /* Processo de archive com mountwait ativado*/
}dsmSendType ;

/*-----------------------------------------------------------------------+
|  Definição de tipo para o parâmetro delType em dsmDeleteObj()          |
+-----------------------------------------------------------------------*/
```

```
typedef enum
{
   dtArchive = 0x00,                      /* Tipo de exclusão de archive */
   dtBackup,                 /* Tipo de exclusão (desativação) de backup */
   dtBackupID                    /* Tipo de exclusão (remoção) de backup */
}dsmDelType ;

/*--------------------------------------------------------------------------+
|  Parâmetro sendType da definição de tipo em dsmSetAccess()                |
+--------------------------------------------------------------------------*/
typedef enum
{
   atBackup = 0x00,                        /* Tipo de processo de backup */
   atArchive                              /* Tipo de processo de archive */
}dsmAccessType;

/*--------------------------------------------------------------------------+
|  Definição de tipo para Versão da API em dsmInit() e dsmQueryApiVersion()|
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t version;       /* Versão da API                    */
   dsUint16_t release;       /* Release da API                   */
   dsUint16_t level;         /* Nível da API                     */
}dsmApiVersion;

/*--------------------------------------------------------------------------+
|  Definição de tipo para Versão da API em dsmInit() e dsmQueryApiVersion()|
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   dsUint16_t version;       /* Versão da API                    */
   dsUint16_t release;       /* Release da API                   */
   dsUint16_t level;         /* Nível da API                     */
   dsUint16_t subLevel;      /* Subnível da API                  */
   dsmBool_t  unicode;       /* Unicode de API?                     */
}dsmApiVersionEx;


#define apiVersionExVer       2

/*--------------------------------------------------------------------------+
|  Definição de tipo para a Versão do Aplicativo em dsmInit()                |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   dsUint16_t   applicationVersion;   /* número da versão do aplicativo */
   dsUint16_t   applicationRelease;   /* número da liberação do aplicativo */
   dsUint16_t   applicationLevel;     /* número do nível do aplicativo */
   dsUint16_t   applicationSubLevel;  /* número do subnível do aplicativo */
} dsmAppVersion;

#define appVersionVer      1

/*--------------------------------------------------------------------------+
|  Definição de tipo para nome do objeto utilizado em BindMC, Send, Delete,|
|  Query                                                                   |
+--------------------------------------------------------------------------*/
typedef struct S_dsmObjName
{
   char       fs[DSM_MAX_FSNAME_LENGTH + 1] ;             /* Nome do espaço no arquivo */
   char       hl[DSM_MAX_HL_LENGTH + 1] ;                /* Nome do nível superior */
   char       ll[DSM_MAX_LL_LENGTH + 1] ;                 /* Nome do nível inferior */
   dsUint8_t  objType; /* para os valores de tipo de objeto, consulte definições acima */
}dsmObjName;


/*--------------------------------------------------------------------------+
```

```
|  Definição de tipo para informações de exclusão de backup em             |
|  dsmDeleteObj()                                                           |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   dsmObjName       *objNameP ;                    /* nome do objeto     */
   dsUint32_t       copyGroup ;               /* grupo de cópias        */
}delBack ;

#define  delBackVersion    1

/*--------------------------------------------------------------------------+
|  Definição de tipo para informações de exclusão de archive em             |
|  dsmDeleteObj()                                                           |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   dsStruct64_t       objId ;                /* ID do objeto           */
}delArch ;

#define  delArchVersion  1

/*--------------------------------------------------------------------------+
|  Definição de tipo para informações de exclusão do ID de Backup em        |
|  dsmDeleteObj()                                                           |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   dsStruct64_t       objId ;                /* ID do objeto           */
}delBackID;

#define  delBackIDVersion  1

/*--------------------------------------------------------------------------+
|  Definição de tipo para informações de exclusão em dsmDeleteObj()         |
+--------------------------------------------------------------------------*/
typedef union
{
   delBack   backInfo ;
   delArch   archInfo ;
   delBackID backIDInfo ;
}dsmDelInfo ;

/*--------------------------------------------------------------------------+
|  Definição de tipo para o parâmetro de Atributo de Objeto em dsmSendObj()|
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   char         owner[DSM_MAX_OWNER_LENGTH + 1]; /* proprietário do objeto */
   dsStruct64_t sizeEstimate; /* Estimativa de tamanho em bytes do objeto */
   dsmBool_t    objCompressed;         /* O objeto já está compactado? */
   dsUint16_t   objInfoLength; /* comprimento das informações dependentes de objeto */
   char         *objInfo;              /* informações dependentes de objeto*/
   char         *mcNameP;      /* nome da classe de gerenciamento para substituição  */
   dsmBool_t    disableDeduplication;         /* não forçar dedup para este objeto    */
   dsmBool_t    useExtObjInfo;                  /* use ext obj info up to 1536 */
}ObjAttr;

#define ObjAttrVersion  4



/*--------------------------------------------------------------------------+
| Definição de tipo para mcBindKey retornado em dsmBindMC()                 |
+--------------------------------------------------------------------------*/
typedef struct
```

```
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char        mcName[DSM_MAX_MC_NAME_LENGTH + 1];
                                        /* Nome de mc ligado ao objeto. */
   dsmBool_t   backup_cg_exists;                     /* Verdadeiro/falso */
   dsmBool_t   archive_cg_exists;                    /* Verdadeiro/falso */
   char        backup_copy_dest[DSM_MAX_CG_DEST_LENGTH + 1];
                                              /* Nome de destino de cópia de backup */
   char        archive_copy_dest[DSM_MAX_CG_DEST_LENGTH + 1];
                                              /* Nome de dest. da cópia de archive */
}mcBindKey;

#define mcBindKeyVersion  1


/*--------------------------------------------------------------------------+
| Definição de tipo para a lista de objetos em dsmBeginGetData()           |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsUint32_t       numObjId ;      /* número de IDs de objetos na lista  */
   ObjID           *objId ;         /* lista de IDs de objetos a restaurar*/
   PartialObjData  *partialObjData; /*lista de inf. de dados do obj. parcial*/
}dsmGetList ;

#define dsmGetListVersion     2  /* padrão se não estiver utilizando dados do Obj. Parcial */
#define dsmGetListPORVersion  3  /* versão se estiver utilizando dados do Obj. Parcial */

/*--------------------------------------------------------------------------+
|  Definição de tipo para DataBlk utilizado para Obter ou Enviar dados     |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsUint32_t bufferLen;            /* Comprimento do buffer transmitido abaixo */
   dsUint32_t numBytes;             /* Número real de bytes lidos         */
                                         /* ou gravados no buffer              */
   char       *bufferPtr;           /* Buffer de dados                   */
   dsUint32_t numBytesCompressed;   /* no envio, bytes compactados        */
   dsUint16_t reserved;                  /* para uso futuro                 */
}DataBlk;

#define DataBlkVersion  3


/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer da Classe de Ger. em dsmBeginQuery() |
+--------------------------------------------------------------------------*/
typedef struct S_qryMCData
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   char         *mcName;                        /* nome da classe de ger. */
                     /* nome único para obter um ou cadeia vazia para obter todos*/
   dsmBool_t    mcDetail;                       /* Deseja detalhes ou não? */
}qryMCData;

#define qryMCDataVersion  1

/*=== valores para RETINIT ===*/
#define ARCH_RETINIT_CREATE 0
#define ARCH_RETINIT_EVENT  1

/*--------------------------------------------------------------------------+
|  Definição de tipo para detalhes do Grupo de Cópias de Archive em        |
|  resposta à MC de Consulta                                               |
+--------------------------------------------------------------------------*/
typedef struct S_archDetailCG
{
   char         cgName[DSM_MAX_CG_NAME_LENGTH + 1]; /* Nome do grupo de cópias */
```

```
   dsUint16_t   frequency;               /* Frequência de cópia (archive) */
   dsUint16_t   retainVers;                                /* Reter versão */
   dsUint8_t    copySer; /* para valores de serialização de cópia, consulte definições */
   dsUint8_t    copyMode;  /* para valores de modo de cópia, consulte definições acima */
   char         destName[DSM_MAX_CG_DEST_LENGTH + 1];        /* Nome do dest. da cópia */
   dsmBool_t    bLanFreeDest;          /* Destino tem caminho sem lan?   */
   dsmBool_t    reserved;                             /* Não utilizado atualmente */
   dsUint8_t    retainInit;                    /* valores possíveis, ver acima */
   dsUint16_t   retainMin;         /* se retInit for EVENT nº de dias     */
   dsmBool_t    bDeduplicate;            /* o destino possui dedup ativada    */
}archDetailCG;

/*--------------------------------------------------------------------------+
|  Definição de tipo para detalhes do Grupo de Cópias de Backup em         |
|  resposta à MC de Consulta                                               |
+--------------------------------------------------------------------------*/
typedef struct S_backupDetailCG
{
   char         cgName[DSM_MAX_CG_NAME_LENGTH + 1]; /* Nome do grupo de cópias */
   dsUint16_t   frequency;                        /* Frequência de backup */
   dsUint16_t   verDataExst;               /* Dados de versões existentes */
   dsUint16_t   verDataDltd;                /* Dados de versões excluídos */
   dsUint16_t   retXtraVers;                      /* Reter versões extras */
   dsUint16_t   retOnlyVers;                     /* Reter somente versões */
   dsUint8_t    copySer; /* para valores de serialização de cópia, consulte definições */
   dsUint8_t    copyMode;  /* para valores de modo de cópia, consulte definições acima */
   char         destName[DSM_MAX_CG_DEST_LENGTH + 1];        /* Nome do dest. da cópia */
   dsmBool_t    bLanFreeDest;          /* Destino tem caminho sem lan?   */
   dsmBool_t    reserved;                             /* Não utilizado atualmente */
   dsmBool_t    bDeduplicate;            /* o destino possui dedup ativada    */
}backupDetailCG;


/*--------------------------------------------------------------------------+
| Definição de tipo para resposta detalhada da Classe de Ger. de Consulta  |
| em dsmGetNextQObj()                                                      |
+--------------------------------------------------------------------------*/
typedef struct S_qryRespMCDetailData
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   char            mcName[DSM_MAX_MC_NAME_LENGTH + 1];      /* nome da mc */
   char            mcDesc[DSM_MAX_MC_DESCR_LENGTH + 1];  /*descrição da mc*/
   archDetailCG    archDet;    /* Detalhes do grupo de cópias de archive */
   backupDetailCG  backupDet;   /* Detalhes do grupo de cópias de backup */
}qryRespMCDetailData;

#define qryRespMCDetailDataVersion  4

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta resumida da Classe de Ger. de Consulta   |
| em dsmGetNextQObj()                                                      |
+--------------------------------------------------------------------------*/
typedef struct S_qryRespMCData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char            mcName[DSM_MAX_MC_NAME_LENGTH + 1];      /* nome da mc */
   char            mcDesc[DSM_MAX_MC_DESCR_LENGTH + 1];  /*descrição da mc*/
}qryRespMCData;

#define qryRespMCDataVersion  1



/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer do Archive em dsmBeginQuery()        |
+--------------------------------------------------------------------------*/
typedef struct S_qryArchiveData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsmObjName     *objName;             /* nome completo de dsm do objeto */
```

```
   char     *owner;                              /* nome do proprietário  */
                     /* para obter os limites máximos de data, consulte as definições acima */
   dsmDate      insDateLowerBound; /* data de inserção de archive de limite inferior */
   dsmDate      insDateUpperBound; /* data de inserção de archive de limite superior */
   dsmDate      expDateLowerBound; /* data de expiração de limite inferior */
   dsmDate      expDateUpperBound; /* data de expiração de limite superior */
   char           *descr;                          /* descrição de archive*/
} qryArchiveData;

#define qryArchiveDataVersion  1



/*=== valores para o campo retentionInitiated ===*/
#define DSM_ARCH_RETINIT_UNKNOWN 0
 /* inicialização da retenção é desconhecida (servidor de nível inferior) */
#define DSM_ARCH_RETINIT_STARTED 1 /* relógio de retenção iniciado        */
#define DSM_ARCH_RETINIT_PENDING 2 /* relógio de retenção não iniciado    */


/*=== Valores para objHeld ===*/
#define DSM_ARCH_HELD_UNKNOWN 0
         /* status de suspensão desconhecido (servidor de nível inferior) */
#define DSM_ARCH_HELD_FALSE   1
         /* objeto NÃO está em um estado de suspensão de exclusão         */
#define DSM_ARCH_HELD_TRUE    2
         /* objeto está em um estado de suspensão de exclusão             */

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta do Archive de Consulta em dsmGetNextQObj()|
+--------------------------------------------------------------------------*/
typedef struct S_qryRespArchiveData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsmObjName      objName; /* qualificador do nomes do espaço no arquivo */
   dsUint32_t     copyGroup;                /* número do grupo de cópias */
   char            mcName[DSM_MAX_MC_NAME_LENGTH + 1];      /* nome da mc */
   char            owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
   dsStruct64_t    objId;                       /* ID exclusivo da cópia */
   dsStruct64_t    reserved;                     /* retrocompatibilidade */
   dsUint8_t       mediaClass;              /* classe de acesso de mídia */
   dsmDate         insDate;              /* data de inserção de archive */
   dsmDate         expDate;           /* data de expiração para o objeto */
   char            descr[DSM_MAX_DESCR_LENGTH + 1]; /* descrição do archive */
   dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
   char            reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
   dsUint160_t     restoreOrderExt;              /* ordem de restauração */
   dsStruct64_t    sizeEstimate; /* estimativa de tamanho armazenada por usuário */
   dsUint8_t       compressType;             /* Sinalizador de compactação */
   dsUint8_t       retentionInitiated; /* objeto aguardando no evento de retenção */
   dsUint8_t       objHeld; /*objeto está em "suspensão" de retenção, consulte valores acima*/
   dsUint8_t       encryptionType;               /* tipo de criptografia*/
   dsmBool_t       clientDeduplicated;             /* obj deduplicado pela API*/
   char            objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
   char            compressAlg[DSM_MAX_COMPRESSTYPE_LENGTH + 1];  /* compression algorithm name */
}qryRespArchiveData;

#define qryRespArchiveDataVersion  7

/*--------------------------------------------------------------------------+
|  Definição de tipo para o parâmetro sendBuff do Archive em dsmSendObj()  |
+--------------------------------------------------------------------------*/
typedef struct S_sndArchiveData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char           *descr;                          /* descrição de archive*/
}sndArchiveData;

#define sndArchiveDataVersion  1
```

```
/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer de Backup em dsmBeginQuery()         |
+--------------------------------------------------------------------------*/
typedef struct S_qryBackupData
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsmObjName     *objName;             /* nome completo de dsm do objeto */
   char     *owner;                             /* nome do proprietário  */
   dsUint8_t      objState;               /* seletor do estado do objeto */
   dsmDate        pitDate; /* Valor da data para restauração de momento específico */
                                /* para obter valores possíveis, consulte as definições acima */
}qryBackupData;

#define qryBackupDataVersion  2

typedef struct
{
  dsUint8_t    reserved1;
  dsStruct64_t reserved2;
} reservedInfo_t;                              /* para utilização futura */

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta de Backup de Consulta em dsmGetNextQObj()|
+--------------------------------------------------------------------------*/
typedef struct S_qryRespBackupData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsmObjName      objName;             /* nome completo de dsm do objeto */
   dsUint32_t     copyGroup;               /* número do grupo de cópias */
   char            mcName[DSM_MAX_MC_NAME_LENGTH + 1];      /* nome da mc */
   char            owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
   dsStruct64_t    objId;                      /* ID exclusivo do objeto */
   dsStruct64_t    reserved;                     /* retrocompatibilidade */
   dsUint8_t       mediaClass;              /* classe de acesso de mídia */
   dsUint8_t       objState;              /* estado do obj., ativo, etc. */
   dsmDate        insDate;                /* data de inserção de backup */
   dsmDate         expDate;           /* data de expiração para o objeto */
   dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
   char            reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
   dsUint160_t     restoreOrderExt;              /* ordem de restauração */
   dsStruct64_t    sizeEstimate; /* estimativa de tamanho armazenada por usuário */
   dsStruct64_t    baseObjId;
   dsUint16_t      baseObjInfolen; /* comprimento de informações dependentes do objeto base*/
   dsUint8_t       baseObjInfo[DSM_MAX_OBJINFO_LENGTH]; /* informações dependentes do objeto base */
   dsUint160_t     baseRestoreOrder;             /* ordem de restauração */
   dsUint32_t      fsID;
   dsUint8_t       compressType;
   dsmBool_t       isGroupLeader;
   dsmBool_t       isOpenGroup;
   dsUint8_t       reserved1;                /* para utilização futura */
   dsmBool_t      reserved2;                 /* para utilização futura */
   dsUint16_t      reserved3;               /* para utilização futura */
   reservedInfo_t  *reserved4;              /* para utilização futura */
   dsUint8_t       encryptionType;              /* tipo de criptografia*/
   dsmBool_t       clientDeduplicated;            /* obj deduplicado pela API*/
   char            objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
   char            compressAlg[DSM_MAX_COMPRESSTYPE_LENGTH + 1];  /* compression algorithm name */
}qryRespBackupData;

#define qryRespBackupDataVersion  8

/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer de Backup Ativo em dsmBeginQuery()   |
|
|  Notas:  Para consulta do backup ativo, somente os campos fs (espaço no
|          arquivo) e objType
|          de objName precisam ser configurados.  objType só pode ser configurado para
|          DSM_OBJ_FILE ou DSM_OBJ_DIRECTORY.  DSM_OBJ_ANY_TYPE não localizará
uma
|          correspondência na consulta.
```

```
+--------------------------------------------------------------------------*/
typedef struct S_qryABackupData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsmObjName      *objName;           /* Somente fs e objtype utilizados */
}qryABackupData;

#define qryABackupDataVersion  1

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta de Backup Ativo de Consulta em           |
| dsmGetNextQObj()                                                         |
+--------------------------------------------------------------------------*/
typedef struct S_qryARespBackupData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsmObjName      objName;             /* nome completo de dsm do objeto */
   dsUint32_t     copyGroup;               /* número do grupo de cópias */
   char            mcName[DSM_MAX_MC_NAME_LENGTH + 1]; /* nome da classe de gerenciamento*/
   char            owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
   dsmDate        insDate;                /* data de inserção de backup */
   dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
   char        reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
   char        objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
}qryARespBackupData;

#define qryARespBackupDataVersion  2

/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer de Backup em dsmBeginQuery()         |
+--------------------------------------------------------------------------*/
typedef struct qryBackupGroups
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint8_t    groupType;
   char         *fsName;
   char           *owner;
   dsStruct64_t groupLeaderObjId;
   dsUint8_t    objType;
   dsmBool_t    noRestoreOrder;
   dsmBool_t    noGroupInfo;
   char         *hl;
}qryBackupGroups;

#define qryBackupGroupsVersion 3


/*--------------------------------------------------------------------------+
|  Definição de tipo para proxynode queryBuffer em dsmBeginQuery()         |
+--------------------------------------------------------------------------*/
typedef struct qryProxyNodeData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char           *targetNodeName;              /* nome do nó de destino  */
}qryProxyNodeData;

#define qryProxyNodeDataVersion  1

/*--------------------------------------------------------------------------+
| Definição de tipo para o parâmetro qryRespProxyNodeData utilizado em     |
| dsmGetNextQObj()                                                         |
+--------------------------------------------------------------------------*/

typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   char            targetNodeName[DSM_MAX_ID_LENGTH+1]; /* nome do nó de destino  */
   char            peerNodeName[DSM_MAX_ID_LENGTH+1];   /* nome do nó de mesmo nível */
   char            hlAddress[DSM_MAX_ID_LENGTH+1];      /* hlAddress de mesmo nível  */
   char            llAddress[DSM_MAX_ID_LENGTH+1];      /* hlAddress de mesmo nível  */
```

```
}qryRespProxyNodeData;

#define qryRespProxyNodeDataVersion  1


/*--------------------------------------------------------------------------+
|  Definição de tipo para atributos de Espaço no Arquivo para WINNT e OS/2 |
+--------------------------------------------------------------------------*/
typedef struct
{
   char         driveLetter ;          /* letra de unidade para espaço no arquivo */
   dsUint16_t   fsInfoLength; /* comprimento de fsInfo utilizado d       */
   char         fsInfo[DSM_MAX_FSINFO_LENGTH];/*dados determinados pelo responsável pela chamada*/
}dsmDosFSAttrib ;

/*--------------------------------------------------------------------------+
|  Definição de tipo para atributos de Espaço no Arquivo do UNIX           |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   fsInfoLength; /* comprimento de fsInfo utilizado d       */
   char         fsInfo[DSM_MAX_FSINFO_LENGTH];/*dados determinados pelo responsável pela chamada*/
}dsmUnixFSAttrib ;

/*--------------------------------------------------------------------------+
|  Definição de tipo para atributos de Espaço no Arquivo do NetWare        |
+--------------------------------------------------------------------------*/
typedef dsmUnixFSAttrib dsmNetwareFSAttrib;

/*--------------------------------------------------------------------------+
|  Definição de tipo para atributos de Espaço no Arquivo em todas as       |
|  chamadas de Espaço no Arquivo                                           |
+--------------------------------------------------------------------------*/
typedef union
{
   dsmNetwareFSAttrib  netwareFSAttr;
   dsmUnixFSAttrib     unixFSAttr ;
   dsmDosFSAttrib      dosFSAttr ;
}dsmFSAttr ;

/*--------------------------------------------------------------------------+
|  Definição de tipo para o parâmetro fsUpd em dsmUpdateFS()
+--------------------------------------------------------------------------*/
typedef struct S_dsmFSUpd
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   char            *fsType ;              /* tipo de espaço no arquivo   */
   dsStruct64_t    occupancy ;         /* estimativa de ocupação         */
   dsStruct64_t    capacity  ;         /* estimativa de capacidade       */
   dsmFSAttr      fsAttr ;         /* atributos específicos da plataforma */
}dsmFSUpd ;

#define dsmFSUpdVersion  1

/*--------------------------------------------------------------------------+
|  Definição de tipo para queryBuffer do Espaço no Arquivo em dsmBeginQuery()             |
+--------------------------------------------------------------------------*/
typedef struct S_qryFSData
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   char           *fsName;                  /* Nome do espaço no arquivo */
}qryFSData;

#define qryFSDataVersion  1

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta do Espaço no Arquivo de Consulta em      |
| dsmGetNextQObj()                                                         |
+--------------------------------------------------------------------------*/
typedef struct S_qryRespFSData
```

```
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char           fsName[DSM_MAX_FSNAME_LENGTH + 1] ; /* Nome do espaço no arquivo */
   char           fsType[DSM_MAX_FSTYPE_LENGTH + 1] ; /* Tipo de espaço no arquivo*/
   dsStruct64_t   occupancy;          /* Estimativa de ocupação em bytes. */
   dsStruct64_t   capacity;                     /* Estimativa de capacidade em bytes. */
   dsmFSAttr      fsAttr ;         /* atributos específicos da plataforma */
   dsmDate        backStartDate;        /* Data do início do backup      */
   dsmDate        backCompleteDate;     /* Data do término do backup     */
   dsmDate        reserved1;                   /* Para utilização futura */
   dsmDate        lastReplStartDate;        /* A última vez que a replicação foi iniciada */
   dsmDate        lastReplCmpltDate;        /* A última vez que a replicação foi concluída */
                                            /*   (pode ter tido uma falha,          */
                                            /*    mas ainda foi concluída)           */
   dsmDate        lastBackOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente */
                                            /*    salvou no servidor   */
   dsmDate        lastArchOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no servidor   */
   dsmDate        lastSpMgOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no servidor   */
   dsmDate        lastBackOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                 */
   dsmDate        lastArchOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                 */
   dsmDate        lastSpMgOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                 */
   dsInt32_t      failOverWriteDelay;       /* Minutos para o cliente aguardar antes de ter permissão */
                                            /* para armazenar neste serv de Replic, Códigos especiais:  */
                                            /* NO_ACCESS(-1), ACCESS_RDONLY (-2)          */
}qryRespFSData;

#define qryRespFSDataVersion 4

/*--------------------------------------------------------------------------+
|  Definição de tipo para o parâmetro regFilespace em dsmRegisterFS()
+--------------------------------------------------------------------------*/
typedef struct S_regFSData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   char           *fsName;         /* Nome do espaço no arquivo           */
   char           *fsType ;        /* Tipo de espaço no arquivo           */
   dsStruct64_t    occupancy;                     /* Estimativa de ocupação em bytes. */
   dsStruct64_t    capacity;                      /* Estimativa de capacidade em bytes. */
   dsmFSAttr      fsAttr ;         /* atributos específicos da plataforma */
}regFSData;

#define regFSDataVersion  1

/*--------------------------------------------------------------------------+
| Definição de tipo para dedupType usado em apisessInfo                     |
+--------------------------------------------------------------------------*/
typedef enum
{
   dedupServerOnly= 0x00,             /* dedup executada apenas no servidor    */
   dedupClientOrServer               /* dedup pode ser executada no cliente ou no servidor */
}dsmDedupType ;

/*--------------------------------------------------------------------------+
 | Definição de tipo para configuração e status de failover
 --------------------------------------------------------------------------*/
typedef enum
{
   failOvrNotConfigured = 0x00,
   failOvrConfigured,
   failOvrConnectedToReplServer
}dsmFailOvrCfgType ;

/*--------------------------------------------------------------------------+
|  Definição de tipo para resposta de informações de sessão em              |
|  dsmQuerySessionInfo()                                                    |
```

```
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
      /*-----------------------------------------------------------------*/
      /*           Informações do servidor                               */
      /*-----------------------------------------------------------------*/
   char           serverHost[DSM_MAX_SERVERNAME_LENGTH+1];
                                  /* Nome do host de rede do servidor DSM */
   dsUint16_t     serverPort;     /* Porta de comunicação do servidor no host */
   dsmDate        serverDate;     /* Data/hora o servidor                 */
   char           serverType[DSM_MAX_SERVERTYPE_LENGTH+1];
                                 /* Plataforma de execução do servidor */
   dsUint16_t     serverVer;        /* Número da versão do servidor       */
   dsUint16_t     serverRel;        /* Número do release do servidor      */
   dsUint16_t     serverLev;        /* Número do nível do servidor        */
   dsUint16_t     serverSubLev;     /* Número do subnível do servidor     */
      /*-----------------------------------------------------------------*/
      /*           Padrões do Cliente                                    */
      /*-----------------------------------------------------------------*/
   char           nodeType[DSM_MAX_PLATFORM_LENGTH+1]; /*tipo de nó/aplicativo*/
   char           fsdelim;                /* Delimitador do espaço no arquivo */
   char           hldelim;                /* Delimitador entre o nível superior e o inferior  */
   dsUint8_t      compression;     /* Sinalizador de compactação          */
   dsUint8_t      archDel;         /* Permissão de exclusão do archive    */
   dsUint8_t      backDel;         /* Permissão de exclusão do backup     */
   dsUint32_t     maxBytesPerTxn;  /* para utilização futura              */
   dsUint16_t     maxObjPerTxn;    /* Máx. de objetos permitidos em um txn*/
      /*-----------------------------------------------------------------*/
      /*           Informações da Sessão                                 */
      /*-----------------------------------------------------------------*/
   char       id[DSM_MAX_ID_LENGTH+1];    /* Nome do nó do ID de conexão  */
   char       owner[DSM_MAX_OWNER_LENGTH+1]; /* Proprietário do registro  */
                                  /*(para plataformas multiusuário)   */
   char       confFile[DSM_PATH_MAX + DSM_NAME_MAX +1];
                                 /* len é dependente de plataforma   */
                                 /* dsInit nome do arquivo de conf. do apl. */
   dsUint8_t  opNoTrace;             /* opção de dsInit - NoTrace = 1    */
      /*-----------------------------------------------------------------*/
      /*           Dados da Política                                     */
      /*-----------------------------------------------------------------*/
   char           domainName[DSM_MAX_DOMAIN_LENGTH+1]; /* Nome do domínio */
   char           policySetName[DSM_MAX_PS_NAME_LENGTH+1];
                                  /* Nome do conjunto de políticas ativo*/
   dsmDate        polActDate;/* Data de ativação do conjunto de políticas */
   char           dfltMCName[DSM_MAX_MC_NAME_LENGTH+1];/* Classe de Ger. Padrão  */
   dsUint16_t     gpBackRetn; /* Retenção de backup no período de tolerância     */
   dsUint16_t     gpArchRetn; /* Retenção de archive no período de tolerância    */
   char           adsmServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* Nome do servidor adsm */
   dsmBool_t      archiveRetentionProtection; /* Está ativado para proteção de retenção do servidor */
   dsStruct64_t   maxBytesPerTxn_64;         /* para uso futuro                */
   dsmBool_t      lanFreeEnabled;            /* a opção sem a lan foi configurada         */
   dsmDedupType   dedupType;                 /* servidor ou clientOrServer       */
   char           accessNode[DSM_MAX_ID_LENGTH+1];    /* como nome do nó         */

   /*-----------------------------------------------------------------*/
   /*           Informações de replicação e failover                  */
   /*-----------------------------------------------------------------*/
   dsmFailOvrCfgType failOverCfgType; /* status de failover */
   char           replServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor de replic */
   char           homeServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor inicial */
   char           replServerHost[DSM_MAX_SERVERNAME_LENGTH+1]; /* Nome do host de rede do servidor DSM  */
   dsInt32_t      replServerPort;                              /* Porta de comunicação do servidor no host       */

}ApiSessInfo;

#define ApiSessInfoVersion  6
```

```
/*--------------------------------------------------------------------------+
| Definição de tipo para resposta de opções de consulta em                  |
| dsmQueryCliOptions()
e dsmQuerySessOptions()                          |
+--------------------------------------------------------------------------*/

typedef struct
{
   char            dsmiDir[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char            dsmiConfig[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char        serverName[DSM_MAX_SERVERNAME_LENGTH+1];
   dsInt16_t   commMethod;
   char        serverAddress[DSM_MAX_SERVER_ADDRESS];
   char        nodeName[DSM_MAX_NODE_LENGTH+1];
   dsmBool_t   compression;
   dsmBool_t   compressalways;
   dsmBool_t   passwordAccess;
}optStruct ;

/*--------------------------------------------------------------------------+
| Definição de tipo para LogType utilizado em logInfo                       |
+--------------------------------------------------------------------------*/
typedef enum
{
   logServer = 0x00,    /* reg. mens. somente no servidor                */
   logLocal,            /* reg. mens. somente no log de erros local      */
   logBoth,                      /* registrar mens no servidor e no log de erro local */
   logNone
}dsmLogType ;

/*--------------------------------------------------------------------------+
| Definição de tipo para o parâmetro logInfo utilizado em dsmLogEvent()     |
+--------------------------------------------------------------------------*/

typedef struct
{
   char               *message;     /* texto da mensagem a ser registrado */
   dsmLogType         logType;    /* tipo de log: local, servidor, ambos */
}logInfo;

/*--------------------------------------------------------------------------+
| Definição de tipo para o parâmetro qryRespAccessData utilizado em         |
| dsmQueryAccess()|                                                         |
+--------------------------------------------------------------------------*/

typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   char              node[DSM_MAX_ID_LENGTH+1];     /* nome do nó         */
   char              owner[DSM_MAX_OWNER_LENGTH+1]; /* proprietário       */
   dsmObjName        objName ;                      /* nome do objeto     */
   dsmAccessType  accessType;                   /* archive ou backup    */
   dsUint32_t     ruleNumber ;                  /* ID da regra de acesso*/
}qryRespAccessData;

#define qryRespAccessDataVersion  1

/*--------------------------------------------------------------------------+
|  Definição de tipo para o parâmetro envSetUp em dsmSetUp()
+--------------------------------------------------------------------------*/
typedef struct S_envSetUp
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   char            dsmiDir[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char            dsmiConfig[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char            dsmiLog[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char            **argv;       /* para o nomes de executáveis argv[0] */
   char            logName[DSM_NAME_MAX +1];
   dsmBool_t      reserved1;                  /* para utilização futura */
   dsmBool_t      reserved2;                  /* para utilização futura */
```

```
}envSetUp;

#define envSetUpVersion  4

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmInitExIn_t                                    |
+-------------------------------------------------------------------------*/
typedef struct dsmInitExIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsmApiVersionEx    *apiVersionExP;
   char               *clientNodeNameP;
   char               *clientOwnerNameP;
   char               *clientPasswordP;
   char               *userNameP;
   char               *userPasswordP;
   char               *applicationTypeP;
   char               *configfile;
   char               *options;
   char               dirDelimiter;
   dsmBool_t          useUnicode;
   dsmBool_t          bCrossPlatform;
   dsmBool_t          bService;
   dsmBool_t          bEncryptKeyEnabled;
   char               *encryptionPasswordP;
   dsmBool_t          useTsmBuffers;
   dsUint8_t          numTsmBuffers;
   dsmAppVersion      *appVersionP;
}dsmInitExIn_t;

#define dsmInitExInVersion 5

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmInitExOut_t                                   |
+-------------------------------------------------------------------------*/
typedef struct dsmInitExOut_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsInt16_t      userNameAuthorities;
   dsInt16_t      infoRC;     /* código de retorno de erro se encontrado */
   char           adsmServerName[DSM_MAX_SERVERNAME_LENGTH+1];
   dsUint16_t     serverVer;        /* Número da versão do servidor       */
   dsUint16_t     serverRel;        /* Número do release do servidor      */
   dsUint16_t     serverLev;        /* Número do nível do servidor        */
   dsUint16_t     serverSubLev;     /* Número do subnível do servidor     */

   dsmBool_t          bIsFailOverMode; /* true se o failover tiver ocorrido */
   char           replServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor de replic */
   char           homeServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor inicial */

}dsmInitExOut_t;

#define dsmInitExOutVersion 3

/*-------------------------------------------------------------------------+
| Definição de tipo para LogType utilizado em logInfo                      |
+-------------------------------------------------------------------------*/
typedef enum
{
   logSevInfo = 0x00,       /* informação ANE4991 */
   logSevWarning,           /* aviso      ANE4992 */
   logSevError,             /* erro       ANE4993 */
   logSevSevere,            /* gravidade  ANE4994 */
   logSevLicense,           /* licença    ANE4995 */
   logSevTryBuy             /* testar Comprar ANE4996 */
}dsmLogSeverity ;


/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmLogExIn_t                                     !
```

```
+--------------------------------------------------------------------------*/
typedef struct dsmLogExIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsmLogSeverity     severity;
   char                appMsgID[8];
   dsmLogType         logType;    /* tipo de log: local, servidor, ambos */
   char                *message;     /* texto da mensagem a ser registrado */
   char                appName[DSM_MAX_PLATFORM_LENGTH];
   char                osPlatform[DSM_MAX_PLATFORM_LENGTH];
   char                appVersion[DSM_MAX_PLATFORM_LENGTH];
}dsmLogExIn_t;

#define dsmLogExInVersion 2

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmlogExOut_t                                     |
+--------------------------------------------------------------------------*/
typedef struct dsmLogExOut_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
}dsmLogExOut_t;

#define dsmLogExOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmRenameIn_t                                     |
+--------------------------------------------------------------------------*/
typedef struct dsmRenameIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t        dsmHandle;            /* identificador para a sessão */
   dsUint8_t      repository;             /* backup ou archive           */
   dsmObjName       *objNameP ;                   /* nome do objeto     */
   char             newHl[DSM_MAX_HL_LENGTH + 1];  /* novo nome de Nível Superior */
   char             newLl[DSM_MAX_LL_LENGTH + 1];  /* novo nome de Nível Inferior */
   dsmBool_t      merge;                          /* associar ao nome existente*/
   ObjID          objId;                  /* objId para Archive          */
}dsmRenameIn_t;

#define dsmRenameInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmRenameOut_t
+--------------------------------------------------------------------------*/
typedef struct dsmRenameOut_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
}dsmRenameOut_t;

#define dsmRenameOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmEndSendObjExIn_t                               |
+--------------------------------------------------------------------------*/
typedef struct dsmEndSendObjExIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsUint32_t        dsmHandle;            /* identificador para a sessão */
}dsmEndSendObjExIn_t;

#define dsmEndSendObjExInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmEndSendObjExOut_t
+--------------------------------------------------------------------------*/
typedef struct dsmEndSendObjExOut_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsStruct64_t       totalBytesSent;     /* total de bytes lidos do apl. */
```

```
   dsmBool_t          objCompressed;     /* foi compactado por objeto     */
   dsStruct64_t       totalCompressSize; /* tamanho total após compactação */
   dsStruct64_t       totalLFBytesSent; /* total de bytes enviados sem Lan */
   dsUint8_t          encryptionType;  /* tipo de criptografia utilizado  */
   dsmBool_t          objDeduplicated;     /* foi o objeto processado para dedup de dados dist. */
   dsStruct64_t       totalDedupSize;      /* tamanho total após dedup */
}dsmEndSendObjExOut_t;

#define dsmEndSendObjExOutVersion 3
/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmGroupHandlerIn_t                              |
+-------------------------------------------------------------------------*/
typedef struct dsmGroupHandlerIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t        dsmHandle;            /* identificador para a sessão */
   dsUint8_t       groupType;        /* tipo de grupo                    */
   dsUint8_t       actionType;       /* tipo de operação do grupo        */
   dsUint8_t       memberType;       /* tipo de membro: Líder ou membro  */
   dsStruct64_t    leaderObjId;      /* OBJID do líder do grupo ao manipular um membro */
   char            *uniqueGroupTagP; /* identificador exclusivo do grupo  */
   dsmObjName      *objNameP ;       /* nome do objeto do líder do grupo  */
   dsmGetList      memberObjList;    /* lista de objetos a remover, designar */
}dsmGroupHandlerIn_t;

#define dsmGroupHandlerInVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmGroupHandlerExOut_t                           |
+-------------------------------------------------------------------------*/
typedef struct dsmGroupHandlerOut_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
}dsmGroupHandlerOut_t;

#define dsmGroupHandlerOutVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmEndTxnExIn_t                                  |
+-------------------------------------------------------------------------*/
typedef struct dsmEndTxnExIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t        dsmHandle;            /* identificador para a sessão */
   dsUint8_t        vote;
}dsmEndTxnExIn_t;

#define dsmEndTxnExInVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmEndTxnExOut_t                                 |
+-------------------------------------------------------------------------*/
typedef struct dsmEndTxnExOut_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint16_t         reason;           /* código de razão                */
   dsStruct64_t       groupLeaderObjId; /* ID do obj. groupLeader no qual foi retornado */
                                        /* DSM_ACTION_OPEN                */
   dsUint8_t      reserved1;         /* utilização futura              */
   dsUint16_t     reserved2;         /* utilização futura              */
}dsmEndTxnExOut_t;

#define dsmEndTxnExOutVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmEndGetDataExIn_t                              |
+-------------------------------------------------------------------------*/
typedef struct dsmEndGetDataExIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
```

```
   dsUint32_t        dsmHandle;            /* identificador para a sessão */
}dsmEndGetDataExIn_t;

#define dsmEndGetDataExInVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para dsmEndGetDataExOut_t                             |
+-------------------------------------------------------------------------*/
typedef struct dsmEndGetDataExOut_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint16_t         reason;           /* código de razão               */
   dsStruct64_t   totalLFBytesRecv;  /* total de bytes sem lan recebidos */
}dsmEndGetDataExOut_t;

#define dsmEndGetDataExOutVersion 1

/*-------------------------------------------------------------------------+
| Definição de tipo para a lista de objetos em dsmRetentionEvent()         |
+-------------------------------------------------------------------------*/
typedef struct dsmObjList
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t     numObjId ;       /* número de IDs de objetos na lista   */
   ObjID          *objId;          /* lista de IDs de objetos a sinalizar */
}dsmObjList_t ;

#define dsmObjlistVersion 1

/*-------------------------------------------------------------------------+
| Definição de tipo eventType utilizada em  dsmRetentionEvent              |
+-------------------------------------------------------------------------*/
typedef enum
{
   eventRetentionActivate = 0x00, /* sinalizar ao servidor que o evento ocorreu*/
   eventHoldObj,                  /* suspender exclusão/expiração do objeto    */
   eventReleaseObj        /* retomar processamento normal de exclusão/expiração*/
}dsmEventType_t;

/*-------------------------------------------------------------------------+
|  Definição de tipo para em dsmRetentionEvent()                           |
+-------------------------------------------------------------------------*/
typedef struct dsmRetentionEventIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
   dsmEventType_t     eventType;                   /* tipo de evento     */
   dsmObjList_t       objList;                     /* ID do objeto       */
}dsmRetentionEventIn_t;

#define dsmRetentionEventInVersion 1

/*-------------------------------------------------------------------------+
|  Definição de tipo para em dsmRetentionEvent()                           |
+-------------------------------------------------------------------------*/
typedef struct dsmRetentionEventOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
}dsmRetentionEventOut_t;

#define dsmRetentionEventOutVersion 1


/*-------------------------------------------------------------------------+
|  Definição de tipo para em dsmRequestBuffer()                            |
+-------------------------------------------------------------------------*/
typedef struct requestBufferIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
```

```
}requestBufferIn_t;

#define requestBufferInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmRequestBuffer()                             |
+--------------------------------------------------------------------------*/
typedef struct requestBufferOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsUint8_t         tsmBufferHandle; /* Identificador para buffer de dados do tsm */
   char               *dataPtr;          /* Endereço no qual gravar dados */
   dsUint32_t       bufferLen;             /* Comp. máx. dos dados a serem gravados */
}requestBufferOut_t;

#define requestBufferOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmReleaseBuffer()                             |
+--------------------------------------------------------------------------*/
typedef struct releaseBufferIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
   dsUint8_t         tsmBufferHandle; /* Identificador para buffer de dados do tsm */
   char               *dataPtr;          /* Endereço no qual gravar dados */
}releaseBufferIn_t;

#define releaseBufferInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmReleaseBuffer()                             |
+--------------------------------------------------------------------------*/
typedef struct releaseBufferOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
}releaseBufferOut_t;

#define releaseBufferOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmGetBufferData()                             |
+--------------------------------------------------------------------------*/
typedef struct getBufferDataIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
}getBufferDataIn_t;

#define getBufferDataInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmGetBufferData()                             |
+--------------------------------------------------------------------------*/
typedef struct getBufferDataOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsUint8_t         tsmBufferHandle; /* Identificador para buffer de dados do tsm */
   char             *dataPtr;        /* Endereço dos dados a serem lidos */
   dsUint32_t       numBytes;    /* Número real de bytes a ler de dataPtr*/
}getBufferDataOut_t;

#define getBufferDataOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmSendBufferData()                            |
+--------------------------------------------------------------------------*/
typedef struct sendBufferDataIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
```

```
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
   dsUint8_t         tsmBufferHandle; /* Identificador para buffer de dados do tsm */
   char              *dataPtr;     /* Endereço dos dados a serem enviados */
   dsUint32_t        numBytes; /* Número real de bytes a enviar de dataPtr*/
}sendBufferDataIn_t;

#define sendBufferDataInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para em dsmSendBufferData()                            |
+--------------------------------------------------------------------------*/
typedef struct sendBufferDataOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
}sendBufferDataOut_t;

#define sendBufferDataOutVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmUpdateObjExIn_t
+--------------------------------------------------------------------------*/
typedef struct dsmUpdateObjExIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t         dsmHandle;            /* Identificador de sessão   */
   dsmSendType     sendType;             /* enviar tipo back/arch */
   char            *descrP;              /* descrição de archive */
   dsmObjName      *objNameP;            /* objName             */
   ObjAttr         *objAttrPtr;          /* atributo           */
   dsUint32_t      objUpdAct;            /* ação de atualização       */
   ObjID           archObjId;            /* objId para archive   */
}dsmUpdateObjExIn_t;

#define dsmUpdateObjExInVersion 1

/*--------------------------------------------------------------------------+
|  Definição de tipo para dsmUpdateObjExOut_t
+--------------------------------------------------------------------------*/
typedef struct dsmUpdateObjExOut_t
{
    dsUint16_t      stVersion;                      /* versão da estrutura*/
}dsmUpdateObjExOut_t;

#define dsmUpdateObjExOutVersion 1

#if (_OPSYS_TYPE == DS_WINNT) && !defined(_WIN64)
#pragma pack()
#endif


#ifdef _MAC
#pragma options align = reset
#endif
#endif /* _H_DSMAPITD */
/*****************************************************************
 * Tivoli Storage Manager                                        *
 * Componente do Cliente da API                                  *
 *                                                               *
 * (C) Copyright IBM Corporation 1993,2010                       *
 *****************************************************************/

/*****************************************************************************
 * Nome do Arquivo de Cabeçalho: tsmapitd.h
 *
 * Ambiente:        ********************************************
 *                  ** Este é um arquivo de origem independente   **
*                  ** de plataforma                              **
 *
 *                  ********************************************
```

```
 *
 * Notas de Design: Este arquivo contém tipos e constantes de dados básicos
 *                  que podem ser incluídos por todos os arquivos de origem do cliente. As constantes
 do arquivo devem ser
*                  configuradas corretamente para a
 máquina ou o sistema
*                  operacional específico no qual o
 software cliente
*                  deve ser executado.
 *
 *                  Definições específicas da plataforma estão incluídas em
*                  dsmapips.h
 *
 * Nome descritivo: Definições para constantes de API do Tivoli Storage
* Manager
 *--------------------------------------------------------------------------*/

#ifndef _H_TSMAPITD
#define _H_TSMAPITD


/*=== configure o alinhamento da estrutura para compactar as estruturas ===*/
#if _OPSYS_TYPE == DS_WINNT
#ifdef _WIN64
#pragma pack(8)
#else
#pragma pack(1)
#endif
#endif

#ifdef _MAC
#pragma options align = packed
#endif

/*================================================================
 Aplicativos Win32 que utilizam a interface do tsm devem
   utilizar o
 sinalizador -DUNICODE durante a compilação.
 ================================================================*/
#if _OPSYS_TYPE == DS_WINNT && !defined(DSMAPILIB)
#ifndef UNICODE
#error "Win32 applications using the TSM interface MUST be compiled with the -DUNICODE flag"
#endif
#endif

/*================================================================
 Aplicativos Mac OS X que utilizam a interface do tsm devem utilizar o
 sinalizador -DUNICODE durante a compilação.
 ================================================================*/
#if _OPSYS_TYPE == DS_MACOS && !defined(DSMAPILIB)
#ifndef UNICODE
#error "Aplicativos Mac OS X que utilizam a interface do TSM DEVEM ser compilados com o sinalizador -DUNICODE"
#endif
#endif

/*--------------------------------------------------------------------------+
 | Definição de tipo para o parâmetro dsmGetType em tsmBeginGetData()       |
 +--------------------------------------------------------------------------*/
typedef enum
{
   gtTsmBackup = 0x00,                    /* Tipo de processo de backup */
   gtTsmArchive                          /* Tipo de processo de archive */
} tsmGetType ;

/*--------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro dsmQueryType em tsmBeginQuery()      |
 +--------------------------------------------------------------------------*/
```

```
typedef enum
{
   qtTsmArchive = 0x00,/* Tipo de consulta de archive                   */
   qtTsmBackup,        /* Tipo de consulta de backup                    */
   qtTsmBackupActive,  /* Consulta rápida para arquivos de backup ativos */
   qtTsmFilespace,     /* Tipo de consulta de espaço no arquivo         */
   qtTsmMC,                              /* Tipo de cons de classe de ger */
    qtTsmReserved1,     /* Utilização futura                             */
    qtTsmReserved2,     /* Utilização futura                             */
    qtTsmReserved3,     /* Utilização futura                             */
    qtTsmReserved4,     /* Utilização futura                             */
   qtTsmBackupGroups,  /* Todos líderes de grupos em um espaço no arquivo específico*/
   qtTsmOpenGroups,    /* Todos membros de grupo associados a um líder   */
    qtTsmReserved5,     /* Utilização futura                             */
   qtTsmProxyNodeAuth, /* Nós para os quais o nó pode efetuar proxy      */
    qtTsmProxyNodePeer,                    /* nós no mesmo nível neste nó de destino */
    qtTsmReserved6,                       /* Utilização futura               */
    qtTsmReserved7,                       /* Utilização futura               */
    qtTsmReserved8                        /* Utilização futura               */
} tsmQueryType ;

/*--------------------------------------------------------------------------+
 |  Parâmetro sendType da definição de tipo em tsmBindMC() e tsmSendObj()   |
 +--------------------------------------------------------------------------*/
typedef enum
{
   stTsmBackup = 0x00,                   /* Tipo de processo de backup */
   stTsmArchive,                         /* Tipo de processo de archive*/
   stTsmBackupMountWait,      /* Processo de backup com mountwait ativado*/
   stTsmArchiveMountWait     /* Processo de archive com mountwait ativado*/
} tsmSendType ;

/*--------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro delType em tsmDeleteObj()            |
 +--------------------------------------------------------------------------*/
typedef enum
{
   dtTsmArchive = 0x00,      /* Tipo de exclusão de archive             */
   dtTsmBackup,              /* Tipo de exclusão (desativação) de backup */
   dtTsmBackupID             /* Tipo de exclusão (remoção) de backup     */
} tsmDelType ;

/*--------------------------------------------------------------------------+
 |  Parâmetro sendType da definição de tipo em tsmSetAccess()               |
 +--------------------------------------------------------------------------*/
typedef enum
{
   atTsmBackup = 0x00,                   /* Tipo de processo de backup */
   atTsmArchive                          /* Tipo de processo de archive*/
}tsmAccessType;

/*--------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro Overwrite em tsmSendObj()            |
 +--------------------------------------------------------------------------*/
typedef enum
{
   owIGNORE = 0x00,
   owYES,
   owNO
}tsmOwType;


/*--------------------------------------------------------------------------+
 |  Definição de tipo para Versão da API em tsmInit() e tsmQueryApiVersion()|
 +--------------------------------------------------------------------------*/
typedef struct
{
```

```
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   dsUint16_t version;       /* Versão da API                    */
   dsUint16_t release;       /* Release da API                   */
   dsUint16_t level;         /* Nível da API                     */
   dsUint16_t subLevel;      /* Subnível da API                  */
   dsmBool_t  unicode;       /* Unicode de API?                     */
} tsmApiVersionEx;

#define tsmApiVersionExVer       2

/*--------------------------------------------------------------------------+
|  Definição de tipo para a Versão do Aplicativo em tsmInit()                |
+--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   dsUint16_t   applicationVersion;   /* número da versão do aplicativo */
   dsUint16_t   applicationRelease;   /* número da liberação do aplicativo */
   dsUint16_t   applicationLevel;     /* número do nível do aplicativo */
   dsUint16_t   applicationSubLevel;  /* número do subnível do aplicativo */
} tsmAppVersion;

#define tsmAppVersionVer      1


/*--------------------------------------------------------------------------+
 |  Definição de tipo para nome do objeto utilizado em BindMC, Send, Delete,|
|  Query                                                                    |
 +--------------------------------------------------------------------------*/

typedef struct tsmObjName
{
   dsChar_t   fs[DSM_MAX_FSNAME_LENGTH + 1] ; /* Nome do espaço no arquivo */
   dsChar_t   hl[DSM_MAX_HL_LENGTH + 1] ;     /* Nome de nível superior */
   dsChar_t   ll[DSM_MAX_LL_LENGTH + 1] ;     /* Nome de nível inferior */
   dsUint8_t  objType; /* para os valores de tipo de objeto, consulte definições acima */
   dsChar_t           dirDelimiter;
} tsmObjName;

/*--------------------------------------------------------------------------+
 |  Definição de tipo para informações de exclusão de backup em             |
|  dsmDeleteObj()                                                           |
 +--------------------------------------------------------------------------*/
typedef struct tsmDelBack
{
   dsUint16_t     stVersion;                     /* versão da estrutura*/
   tsmObjName       *objNameP ;               /* nome do objeto         */
   dsUint32_t       copyGroup ;               /* grupo de cópias        */
} tsmDelBack ;

#define  tsmDelBackVersion    1

/*--------------------------------------------------------------------------+
 |  Definição de tipo para informações de exclusão de archive em            |
|  dsmDeleteObj()                                                           |
 +--------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;                     /* versão da estrutura*/
   dsStruct64_t       objId ;                /* ID do objeto           */
} tsmDelArch ;

#define  tsmDelArchVersion  1

/*--------------------------------------------------------------------------+
 |  Definição de tipo para informações de exclusão do ID de Backup em       |
|  dsmDeleteObj()                                                           |
```

```
 +---------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsStruct64_t       objId ;                 /* ID do objeto           */
} tsmDelBackID;

#define  tsmDelBackIDVersion  1


/*-----------------------------------------------------------------------+
 |  Definição de tipo para informações de exclusão em dsmDeleteObj()     |
 +-----------------------------------------------------------------------*/
typedef union
{
   tsmDelBack   backInfo ;
   tsmDelArch   archInfo ;
   tsmDelBackID backIDInfo;
} tsmDelInfo ;

/*-----------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro de Atributo de Objeto em dsmSendObj()|
 +-----------------------------------------------------------------------*/
typedef struct tsmObjAttr
{
   dsUint16_t   stVersion;                       /* versão da estrutura   */
   dsChar_t     owner[DSM_MAX_OWNER_LENGTH + 1]; /* proprietário do objeto*/
   dsStruct64_t sizeEstimate; /* Estimativa de tamanho em bytes do objeto */
   dsmBool_t    objCompressed;         /* O objeto já está compactado?     */
   dsUint16_t   objInfoLength; /* comprimento das informações dependentes de objeto */
   char         *objInfo;      /* buffer de bytes de informações dependentes do objeto */
   dsChar_t     *mcNameP;      /* nome da classe de gerenciamento para substituição*/
   tsmOwType    reserved1;                     /* para utilização futura */
   tsmOwType    reserved2;                     /* para utilização futura */
   dsmBool_t    disableDeduplication;         /* não forçar dedup para este objeto    */
   dsmBool_t    useExtObjInfo;             /* use ext objinfo up to 1536        */
} tsmObjAttr;

#define tsmObjAttrVersion 5

/*-----------------------------------------------------------------------+
 | Definição de tipo para mcBindKey retornado em dsmBindMC()              |
 +-----------------------------------------------------------------------*/
typedef struct tsmMcBindKey
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsChar_t    mcName[DSM_MAX_MC_NAME_LENGTH + 1];
   /* Nome de mc ligado ao objeto. */
   dsmBool_t   backup_cg_exists;                     /* Verdadeiro/falso */
   dsmBool_t   archive_cg_exists;                    /* Verdadeiro/falso */
   dsChar_t    backup_copy_dest[DSM_MAX_CG_DEST_LENGTH + 1];
   /* Nome de destino de cópia de backup */
   dsChar_t    archive_copy_dest[DSM_MAX_CG_DEST_LENGTH + 1];
   /* Nome de dest. da cópia de archive */
} tsmMcBindKey;

#define tsmMcBindKeyVersion  1


/*-----------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer da Classe de Ger. em dsmBeginQuery() |
 +-----------------------------------------------------------------------*/
typedef struct tsmQryMCData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsChar_t         *mcName;                  /* nome da classe de ger. */
   /* nome único para obter um ou cadeia vazia para obter todos*/
```

```
   dsmBool_t    mcDetail;                          /* Deseja detalhes ou não? */
} tsmQryMCData;

#define tsmQryMCDataVersion  1


/*--------------------------------------------------------------------------+
 |  Definição de tipo para detalhes do Grupo de Cópias de Archive em        |
|  resposta à MC de Consulta                                                |
 +--------------------------------------------------------------------------*/
typedef struct tsmArchDetailCG
{
   dsChar_t     cgName[DSM_MAX_CG_NAME_LENGTH + 1]; /* Nome do grupo de cópias */
   dsUint16_t   frequency;               /* Freqüência de cópia (archive) */
   dsUint16_t   retainVers;                                /* Reter versão */
   dsUint8_t    copySer; /* para valores de serialização de cópia, consulte definições */
   dsUint8_t    copyMode;  /* para valores de modo de cópia, consulte definições acima */
   dsChar_t     destName[DSM_MAX_CG_DEST_LENGTH + 1]; /* Nome do dest. da cópia */
   dsmBool_t    bLanFreeDest;           /* Destino tem caminho sem lan?    */
   dsmBool_t    reserved;                             /* Não utilizado atualmente */
   dsUint8_t    retainInit;        /* valores possíveis, ver dsmapitd.h   */
   dsUint16_t   retainMin;         /* se retInit for EVENT nº de dias     */
   dsmBool_t    bDeduplicate;             /* o destino possui dedup ativada  */
}tsmArchDetailCG;

/*--------------------------------------------------------------------------+
 |  Definição de tipo para detalhes do Grupo de Cópias de Backup em         |
|  resposta à MC de Consulta                                                |
 +--------------------------------------------------------------------------*/
typedef struct tsmBackupDetailCG
{
   dsChar_t     cgName[DSM_MAX_CG_NAME_LENGTH + 1]; /* Nome do grupo de cópias */
   dsUint16_t   frequency;                        /* Frequência de backup */
   dsUint16_t   verDataExst;               /* Dados de versões existentes */
   dsUint16_t   verDataDltd;                /* Dados de versões excluídos */
   dsUint16_t   retXtraVers;                      /* Reter versões extras */
   dsUint16_t   retOnlyVers;                     /* Reter somente versões */
   dsUint8_t    copySer; /* para valores de serialização de cópia, consulte definições */
   dsUint8_t    copyMode;  /* para valores de modo de cópia, consulte definições acima */
   dsChar_t     destName[DSM_MAX_CG_DEST_LENGTH + 1]; /* Nome do dest. da cópia */
   dsmBool_t    bLanFreeDest;           /* Destino tem caminho sem lan?    */
   dsmBool_t    reserved;                             /* Não utilizado atualmente */
   dsmBool_t    bDeduplicate;             /* o destino possui dedup ativada  */
}tsmBackupDetailCG;


/*--------------------------------------------------------------------------+
 | Definição de tipo para resposta detalhada da Classe de Ger. de Consulta  |
| em dsmGetNextQObj()                                                       |
 +--------------------------------------------------------------------------*/
typedef struct tsmQryRespMCDetailData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsChar_t        mcName[DSM_MAX_MC_NAME_LENGTH + 1];     /* nome da mc */
   dsChar_t        mcDesc[DSM_MAX_MC_DESCR_LENGTH + 1]; /*descrição da mc*/
   archDetailCG    archDet;    /* Detalhes do grupo de cópias de archive */
   backupDetailCG  backupDet;   /* Detalhes do grupo de cópias de backup */
} tsmQryRespMCDetailData;

#define tsmQryRespMCDetailDataVersion  4

/*--------------------------------------------------------------------------+
 | Definição de tipo para resposta resumida da Classe de Ger. de Consulta   |
| em dsmGetNextQObj()                                                       |
 +--------------------------------------------------------------------------*/
typedef struct tsmQryRespMCData
{
```

```
   dsUint16_t         stVersion;        /* versão da estrutura           */
   dsChar_t        mcName[DSM_MAX_MC_NAME_LENGTH + 1];     /* nome da mc */
   dsChar_t       mcDesc[DSM_MAX_MC_DESCR_LENGTH + 1];  /*descrição da mc*/
}tsmQryRespMCData;

#define tsmQryRespMCDataVersion  1

/*--------------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer do Archive em tsmBeginQuery()        |
 +--------------------------------------------------------------------------*/
typedef struct tsmQryArchiveData
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   tsmObjName    *objName;             /* nome completo de dsm do objeto */
   dsChar_t      *owner;                         /* nome do proprietário */
   /* para obter os limites máximos de data, consulte as definições acima */
   dsmDate      insDateLowerBound; /* data de inserção de archive de limite inferior */
   dsmDate      insDateUpperBound; /* data de inserção de archive de limite superior */
   dsmDate      expDateLowerBound; /* data de expiração de limite inferior */
   dsmDate      expDateUpperBound; /* data de expiração de limite superior */
   dsChar_t         *descr;                       /* descrição de archive*/
} tsmQryArchiveData;

#define tsmQryArchiveDataVersion  1

/*--------------------------------------------------------------------------+
| Definição de tipo para resposta do Archive de Consulta em dsmGetNextQObj()|
+--------------------------------------------------------------------------*/
typedef struct tsmQryRespArchiveData
{
    dsUint16_t         stVersion;                   /* versão da estrutura*/
    tsmObjName      objName; /* Qualificador do nome do espaço no arquivo */
    dsUint32_t     copyGroup;               /* número do grupo de cópias */
    dsChar_t        mcName[DSM_MAX_MC_NAME_LENGTH + 1];     /* nome da mc */
    dsChar_t       owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
    dsStruct64_t    objId;                       /* ID exclusivo da cópia */
    dsStruct64_t    reserved;                     /* retrocompatibilidade */
    dsUint8_t       mediaClass;             /* classe de acesso de mídia */
    dsmDate         insDate;               /* data de inserção de archive */
    dsmDate         expDate;           /* data de expiração para o objeto */
    dsChar_t        descr[DSM_MAX_DESCR_LENGTH + 1]; /* descrição do archive */
    dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
    dsUint8_t       reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
    dsUint160_t     restoreOrderExt;             /* ordem de restauração */
    dsStruct64_t    sizeEstimate; /* estimativa de tamanho armazenada por usuário */
    dsUint8_t       compressType;          /* Sinalizador de compactação */
    dsUint8_t retentionInitiated; /* objeto aguardando no evento de retenção */
    dsUint8_t       objHeld; /* objeto em "suspensão" consulte dsmapitd.h para obter valores */
    dsUint8_t       encryptionType;              /* tipo de criptografia*/
    dsmBool_t       clientDeduplicated;            /* obj deduplicado pela API*/
    dsUint8_t       objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
    dsChar_t        compressAlg[DSM_MAX_COMPRESSTYPE_LENGTH + 1];  /* compression algorithm name */
} tsmQryRespArchiveData;

#define tsmQryRespArchiveDataVersion  7

/*--------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro sendBuff do Archive em dsmSendObj()  |
 +--------------------------------------------------------------------------*/
typedef struct tsmSndArchiveData
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsChar_t        *descr;                       /* descrição de archive*/
} tsmSndArchiveData;

#define tsmSndArchiveDataVersion  1
```

```
/*--------------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer de Backup em dsmBeginQuery()         |
 +--------------------------------------------------------------------------*/
typedef struct tsmQryBackupData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   tsmObjName     *objName;            /* nome completo de dsm do objeto */
   dsChar_t      *owner;                         /* nome do proprietário */
   dsUint8_t      objState;               /* seletor do estado do objeto */
   dsmDate        pitDate; /* Valor da data para restauração de momento específico */
   /* para obter valores possíveis, consulte as definições acima */
    dsUint32_t  reserved1;
    dsUint32_t  reserved2;
} tsmQryBackupData;

#define tsmQryBackupDataVersion  3

/*--------------------------------------------------------------------------+
 | Definição de tipo para resposta de Backup de Consulta em dsmGetNextQObj()|
 +--------------------------------------------------------------------------*/
typedef struct tsmQryRespBackupData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   tsmObjName     objName;             /* nome completo de dsm do objeto */
   dsUint32_t     copyGroup;                /* número do grupo de cópias */
   dsChar_t        mcName[DSM_MAX_MC_NAME_LENGTH + 1];     /* nome da mc */
   dsChar_t       owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
   dsStruct64_t    objId;                       /* ID exclusivo do objeto */
   dsStruct64_t    reserved;                      /* retrocompatibilidade */
   dsUint8_t       mediaClass;              /* classe de acesso de mídia */
   dsUint8_t       objState;              /* estado do obj., ativo, etc. */
   dsmDate        insDate;                /* data de inserção de backup */
   dsmDate         expDate;           /* data de expiração para o objeto */
   dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
   dsUint8_t       reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
   dsUint160_t     restoreOrderExt;              /* ordem de restauração */
   dsStruct64_t    sizeEstimate; /* estimativa de tamanho armazenada por usuário */
   dsStruct64_t    baseObjId;
   dsUint16_t      baseObjInfolen; /* comprimento de informações dependentes do objeto base*/
   dsUint8_t       baseObjInfo[DSM_MAX_OBJINFO_LENGTH]; /* informações dependentes do objeto base */
   dsUint160_t     baseRestoreOrder;             /* ordem de restauração */
   dsUint32_t      fsID;
   dsUint8_t       compressType;
   dsmBool_t       isGroupLeader;
   dsmBool_t       isOpenGroup;
   dsUint8_t       reserved1;                /* para utilização futura */
   dsmBool_t      reserved2;                 /* para utilização futura */
   dsUint16_t      reserved3;               /* para utilização futura */
   reservedInfo_t  *reserved4;              /* para utilização futura */
   dsUint8_t       encryptionType;             /* tipo de criptografia*/
   dsmBool_t       clientDeduplicated;           /* obj deduplicado pela API*/
   dsUint8_t       objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
   dsChar_t        compressAlg[DSM_MAX_COMPRESSTYPE_LENGTH + 1];  /* compression algorithm name */
} tsmQryRespBackupData;

#define tsmQryRespBackupDataVersion  8
/*--------------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer de Backup Ativo em dsmBeginQuery()   |
 |                                                                          |
 |  Notas:  Para consulta do backup ativo, somente os campos fs (espaço no
|          arquivo) e objType
 |          de objName precisam ser configurados.  objType só pode ser configurado para
 |          DSM_OBJ_FILE ou DSM_OBJ_DIRECTORY.  DSM_OBJ_ANY_TYPE não localizará
 uma
|          correspondência na consulta.
 +--------------------------------------------------------------------------*/
typedef struct tsmQryABackupData
```

```
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   tsmObjName      *objName;          /* Somente fs e objtype utilizados */
} tsmQryABackupData;

#define tsmQryABackupDataVersion  1

/*-------------------------------------------------------------------------+
 | Definição de tipo para resposta de Backup Ativo de Consulta em          |
| dsmGetNextQObj()                                                         |
 +-------------------------------------------------------------------------*/
typedef struct tsmQryARespBackupData
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   tsmObjName     objName;             /* nome completo de dsm do objeto */
   dsUint32_t     copyGroup;                /* número do grupo de cópias */
   dsChar_t       mcName[DSM_MAX_MC_NAME_LENGTH + 1]; /* nome da classe de gerenciamento */
   dsChar_t       owner[DSM_MAX_OWNER_LENGTH + 1]; /* nome do proprietário */
   dsmDate        insDate;                 /* data de inserção de backup */
   dsUint16_t     objInfolen; /* comprimento das informações dependentes de objeto*/
   dsUint8_t   reservedObjInfo[DSM_MAX_OBJINFO_LENGTH];  /*object-dependent info */
   dsUint8_t  objInfo[DSM_MAX_EXT_OBJINFO_LENGTH];  /*object-dependent info */
} tsmQryARespBackupData;

#define tsmQryARespBackupDataVersion  2

/*-------------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer de Backup em dsmBeginQuery()        |
 +-------------------------------------------------------------------------*/
typedef struct tsmQryBackupGroups
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint8_t    groupType;
   dsChar_t     *fsName;
   dsChar_t     *owner;
   dsStruct64_t groupLeaderObjId;
   dsUint8_t    objType;
   dsUint32_t   reserved1;
   dsUint32_t   reserverd2;
   dsmBool_t    noRestoreOrder;
   dsmBool_t    noGroupInfo;
   dsChar_t     *hl;
} tsmQryBackupGroups;

#define tsmQryBackupGroupsVersion 4

/*-------------------------------------------------------------------------+
 |  Definição de tipo para proxynode queryBuffer em tsmBeginQuery()        |
 +-------------------------------------------------------------------------*/
typedef struct tsmQryProxyNodeData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsChar_t       *targetNodeName;             /* nome do nó de destino  */
}tsmQryProxyNodeData;

#define tsmQryProxyNodeDataVersion  1

/*-------------------------------------------------------------------------+
 | Definição de tipo para o parâmetro qryRespProxyNodeData utilizado em    |
| tsmGetNextQObj()                                                         |
 +-------------------------------------------------------------------------*/

typedef struct tsmQryRespProxyNodeData
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   dsChar_t            targetNodeName[DSM_MAX_ID_LENGTH+1]; /* nome do nó de destino  */
   dsChar_t            peerNodeName[DSM_MAX_ID_LENGTH+1];   /* nome do nó de mesmo nível */
```

```
   dsChar_t            hlAddress[DSM_MAX_ID_LENGTH+1];      /* hlAddress de mesmo nível   */
   dsChar_t            llAddress[DSM_MAX_ID_LENGTH+1];      /* llAddress de mesmo nível   */
}tsmQryRespProxyNodeData;

#define tsmQryRespProxyNodeDataVersion  1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para atributos de Espaço no Arquivo para WINNT e OS/2 |
 +-------------------------------------------------------------------------*/
typedef struct tsmDosFSAttrib
{
   osChar_t     driveLetter ; /* letra de unidade para espaço no arquivo */
   dsUint16_t   fsInfoLength; /* comprimento de fsInfo utilizado d       */
   osChar_t     fsInfo[DSM_MAX_FSINFO_LENGTH];/*dados determinados pelo responsável pela chamada*/
} tsmDosFSAttrib ;

/*-------------------------------------------------------------------------+
 |  Definição de tipo para atributos de Espaço no Arquivo do UNIX           |
 +-------------------------------------------------------------------------*/
typedef struct tsmUnixFSAttrib
{
   dsUint16_t   fsInfoLength; /* comprimento de fsInfo utilizado d       */
   osChar_t     fsInfo[DSM_MAX_FSINFO_LENGTH];/*dados determinados pelo responsável pela chamada*/
} tsmUnixFSAttrib ;

/*-------------------------------------------------------------------------+
 |  Definição de tipo para atributos de Espaço no Arquivo do NetWare        |
 +-------------------------------------------------------------------------*/
typedef tsmUnixFSAttrib tsmNetwareFSAttrib;

/*-------------------------------------------------------------------------+
 |  Definição de tipo para atributos de Espaço no Arquivo em todas as       |
|  chamadas de Espaço no Arquivo                                            |
 +-------------------------------------------------------------------------*/
typedef union
{
   tsmNetwareFSAttrib  netwareFSAttr;
   tsmUnixFSAttrib     unixFSAttr ;
   tsmDosFSAttrib      dosFSAttr ;
} tsmFSAttr ;

/*-------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro fsUpd em dsmUpdateFS()
 +-------------------------------------------------------------------------*/
typedef struct   tsmFSUpd
{
   dsUint16_t     stVersion;        /* versão da estrutura                */
   dsChar_t        *fsType ;           /* tipo de espaço no arquivo      */
   dsStruct64_t    occupancy ;         /* estimativa de ocupação         */
   dsStruct64_t    capacity  ;         /* estimativa de capacidade       */
   tsmFSAttr      fsAttr ;        /* atributos específicos da plataforma */
} tsmFSUpd ;

#define tsmFSUpdVersion  1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para queryBuffer do Espaço no Arquivo em dsmBeginQuery()             |
 +-------------------------------------------------------------------------*/
typedef struct tsmQryFSData
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsChar_t        *fsName;                 /* Nome do espaço no arquivo */
} tsmQryFSData;

#define tsmQryFSDataVersion  1

/*-------------------------------------------------------------------------+
```

```
 | Definição de tipo para resposta do Espaço no Arquivo de Consulta em      |
| dsmGetNextQObj()                                                          |
 +-------------------------------------------------------------------------*/
typedef struct tsmQryRespFSData
{
   dsUint16_t        stVersion;                    /* versão da estrutura*/
   dsChar_t       fsName[DSM_MAX_FSNAME_LENGTH + 1] ; /* Nome do espaço no arquivo */
   dsChar_t       fsType[DSM_MAX_FSTYPE_LENGTH + 1] ; /* Tipo de espaço no arquivo*/
   dsStruct64_t    occupancy;                     /* Estimativa de ocupação em bytes.             */
   dsStruct64_t    capacity;                      /* Estimativa de capacidade em bytes.               */
   tsmFSAttr      fsAttr ;        /* atributos específicos da plataforma */
   dsmDate        backStartDate;        /* Data do início do backup      */
   dsmDate        backCompleteDate;     /* Data do término do backup     */
    dsmDate        reserved1;                  /* Para utilização futura */
   dsmBool_t      bIsUnicode;
   dsUint32_t      fsID;
   dsmDate        lastReplStartDate;        /* A última vez que a replicação foi iniciada */
   dsmDate        lastReplCmpltDate;        /* A última vez que a replicação foi concluída */
                                            /*   (pode ter tido uma falha,          */
                                            /*    mas ainda foi concluída)            */
   dsmDate        lastBackOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente */
                                            /*    salvou no servidor   */
   dsmDate        lastArchOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no servidor   */
   dsmDate        lastSpMgOpDateFromServer; /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no servidor   */
   dsmDate        lastBackOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                */
   dsmDate        lastArchOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                */
   dsmDate        lastSpMgOpDateFromLocal;  /* O último registro de data e hora de armazenamento que o cliente  */
                                            /*    salvou no Local                */
   dsInt32_t      failOverWriteDelay;       /* Minutos para o cliente aguardar antes de ter permissão */
                                            /* para armazenar neste serv de Replic, Códigos especiais:  */
                                            /* NO_ACCESS(-1), ACCESS_RDONLY (-2)           */
} tsmQryRespFSData;

#define tsmQryRespFSDataVersion 5

/*-------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro regFilespace em dsmRegisterFS()
 +-------------------------------------------------------------------------*/
typedef struct tsmRegFSData
{
   dsUint16_t        stVersion;                    /* versão da estrutura*/
   dsChar_t       *fsName;                 /* Nome do espaço no arquivo */
   dsChar_t       *fsType ;                /* Tipo de espaço no arquivo */
   dsStruct64_t   occupancy;                     /* Estimativa de ocupação em bytes. */
   dsStruct64_t    capacity;                      /* Estimativa de capacidade em bytes. */
   tsmFSAttr      fsAttr ;        /* atributos específicos da plataforma */
} tsmRegFSData;

#define tsmRegFSDataVersion  1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para resposta de informações de sessão em             |
|  dsmQuerySessionInfo()                                                    |
 +-------------------------------------------------------------------------*/
typedef struct
{
   dsUint16_t   stVersion;                      /* versão da estrutura   */
   /*-----------------------------------------------------------------*/
   /*           Informações do servidor                               */
   /*-----------------------------------------------------------------*/
   dsChar_t       serverHost[DSM_MAX_SERVERNAME_LENGTH+1];
   /* Nome do host de rede do servidor DSM */
   dsUint16_t     serverPort;     /* Porta de comunicação do servidor no host */
```

```
   dsmDate        serverDate;     /* Data/hora o servidor                */
   dsChar_t       serverType[DSM_MAX_SERVERTYPE_LENGTH+1];
   /* Plataforma de execução do servidor */
   dsUint16_t     serverVer;        /* Número da versão do servidor       */
   dsUint16_t     serverRel;        /* Número do release do servidor      */
   dsUint16_t     serverLev;        /* Número do nível do servidor        */
   dsUint16_t     serverSubLev;     /* Número do subnível do servidor     */
   /*--------------------------------------------------------------------*/
   /*           Padrões do Cliente                                       */
   /*--------------------------------------------------------------------*/
   dsChar_t       nodeType[DSM_MAX_PLATFORM_LENGTH+1]; /*tipo de nó/aplicativo*/
   dsChar_t       fsdelim;         /* Delimitador do espaço no arquivo    */
   dsChar_t       hldelim;         /* Delimitador entre o nível superior e o inferior  */
   dsUint8_t      compression;     /* Sinalizador de compactação          */
   dsUint8_t      archDel;         /* Permissão de exclusão do archive    */
   dsUint8_t      backDel;         /* Permissão de exclusão do backup     */
   dsUint32_t     maxBytesPerTxn;  /* para utilização futura              */
   dsUint16_t     maxObjPerTxn;    /* Máx. de objetos permitidos em um txn*/
   /*--------------------------------------------------------------------*/
   /*           Informações da Sessão                                    */
   /*--------------------------------------------------------------------*/
   dsChar_t       id[DSM_MAX_ID_LENGTH+1]; /* Nome do nó do ID de conexão */
   dsChar_t       owner[DSM_MAX_OWNER_LENGTH+1]; /* Proprietário do registro*/
   /*(para plataformas multiusuário)   */
   dsChar_t       confFile[DSM_PATH_MAX + DSM_NAME_MAX +1];
   /* len é dependente de plataforma   */
   /* dsInit nome do arquivo de conf. do apl. */
   dsUint8_t  opNoTrace;             /* opção de dsInit - NoTrace = 1    */
   /*--------------------------------------------------------------------*/
   /*           Dados da Política                                        */
   /*--------------------------------------------------------------------*/
   dsChar_t       domainName[DSM_MAX_DOMAIN_LENGTH+1]; /* Nome do domínio */
   dsChar_t       policySetName[DSM_MAX_PS_NAME_LENGTH+1];
   /* Nome do conjunto de políticas ativo*/
   dsmDate        polActDate;/* Data de ativação do conjunto de políticas */
   dsChar_t       dfltMCName[DSM_MAX_MC_NAME_LENGTH+1];/* Classe de Ger. Padrão  */
   dsUint16_t     gpBackRetn; /* Retenção de backup no período de tolerância     */
   dsUint16_t     gpArchRetn; /* Retenção de archive no período de tolerância    */
   dsChar_t           adsmServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor adsm */
  dsmBool_t      archiveRetentionProtection; /* Está ativado para proteção de retenção do servidor */
   dsUint64_t     maxBytesPerTxn_64;         /* para uso futuro                 */
   dsmBool_t      lanFreeEnabled;            /* a opção sem a lan foi configurada         */
   dsmDedupType   dedupType;                 /* servidor ou clientOrServer      */
   dsChar_t       accessNode[DSM_MAX_ID_LENGTH+1];    /* como nome do nó         */

   /*--------------------------------------------------------------------*/
   /*           Informações de replicação e failover                  */
   /*--------------------------------------------------------------------*/
   dsmFailOvrCfgType failOverCfgType; /* status de failover */
   dsChar_t        replServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor de replic */
   dsChar_t        homeServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor inicial */
   dsChar_t        replServerHost[DSM_MAX_SERVERNAME_LENGTH+1]; /* Nome do host de rede do servidor DSM  */
   dsInt32_t      replServerPort;                              /* Porta de comunicação do servidor no host       */

} tsmApiSessInfo;

#define tsmApiSessInfoVersion  6

/*--------------------------------------------------------------------------+
 | Definição de tipo para resposta de opções de consulta em                 |
| dsmQueryCliOptions()
 e dsmQuerySessOptions()                              |
 +--------------------------------------------------------------------------*/

typedef struct
{
   dsUint16_t  stVersion;
```

```
   dsChar_t        dsmiDir[DSM_PATH_MAX + DSM_NAME_MAX +1];
   dsChar_t        dsmiConfig[DSM_PATH_MAX + DSM_NAME_MAX +1];
   dsChar_t    serverName[DSM_MAX_SERVERNAME_LENGTH+1];
   dsInt16_t   commMethod;
   dsChar_t    serverAddress[DSM_MAX_SERVER_ADDRESS];
   dsChar_t    nodeName[DSM_MAX_NODE_LENGTH+1];
   dsmBool_t   compression;
   dsmBool_t   compressalways;
   dsmBool_t   passwordAccess;
}tsmOptStruct ;

#define tsmOptStructVersion  1

/*------------------------------------------------------------------------+
 | Definição de tipo para o parâmetro qryRespAccessData utilizado em      |
| dsmQueryAccess()|                                                       |
 +------------------------------------------------------------------------*/

typedef struct
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
   dsChar_t       node[DSM_MAX_ID_LENGTH+1];       /* nome do nó         */
   dsChar_t       owner[DSM_MAX_OWNER_LENGTH + 1]; /* proprietário       */
   tsmObjName     objName ;                        /* nome do objeto     */
   dsmAccessType  accessType;                      /* archive ou backup  */
   dsUint32_t     ruleNumber ;                     /* ID da regra de acesso*/

}tsmQryRespAccessData;

#define tsmQryRespAccessDataVersion  1

/*------------------------------------------------------------------------+
 |  Definição de tipo para o parâmetro envSetUp em dsmSetUp()
 +------------------------------------------------------------------------*/
typedef struct tsmEnvSetUp
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsChar_t        dsmiDir[DSM_PATH_MAX + DSM_NAME_MAX +1];
   dsChar_t        dsmiConfig[DSM_PATH_MAX + DSM_NAME_MAX +1];
   dsChar_t        dsmiLog[DSM_PATH_MAX + DSM_NAME_MAX +1];
   char            **argv;       /* para o nomes de executáveis argv[0] */
   dsChar_t        logName[DSM_NAME_MAX +1];
   dsmBool_t      reserved1;                   /* para utilização futura */
   dsmBool_t      reserved2;                   /* para utilização futura */
} tsmEnvSetUp;

#define tsmEnvSetUpVersion  4

/*------------------------------------------------------------------------+
 |  Definição de tipo para dsmInitExIn_t                                  |
 +------------------------------------------------------------------------*/
typedef struct tsmInitExIn_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   tsmApiVersionEx    *apiVersionExP;
   dsChar_t           *clientNodeNameP;
   dsChar_t           *clientOwnerNameP;
   dsChar_t           *clientPasswordP;
   dsChar_t           *userNameP;
   dsChar_t           *userPasswordP;
   dsChar_t           *applicationTypeP;
   dsChar_t           *configfile;
   dsChar_t           *options;
   dsChar_t           dirDelimiter;
   dsmBool_t          useUnicode;
   dsmBool_t          bCrossPlatform;
   dsmBool_t          bService;
```

```
   dsmBool_t          bEncryptKeyEnabled;
   dsChar_t           *encryptionPasswordP;
   dsmBool_t          useTsmBuffers;
   dsUint8_t          numTsmBuffers;
   tsmAppVersion      appVersionP;
} tsmInitExIn_t;

#define tsmInitExInVersion 5

/*------------------------------------------------------------------------+
 |  Definição de tipo para dsmInitExOut_t                                 |
 +------------------------------------------------------------------------*/
typedef struct tsmInitExOut_t
{
   dsUint16_t         stVersion;                  /* versão da estrutura*/
   dsInt16_t      userNameAuthorities;
   dsInt16_t      infoRC;     /* código de retorno de erro se encontrado */
   /* nome do servidor adsm            */
   dsChar_t       adsmServerName[DSM_MAX_SERVERNAME_LENGTH+1];
   dsUint16_t     serverVer;        /* Número da versão do servidor       */
   dsUint16_t     serverRel;        /* Número do release do servidor      */
   dsUint16_t     serverLev;        /* Número do nível do servidor        */
   dsUint16_t     serverSubLev;     /* Número do subnível do servidor     */
   dsmBool_t          bIsFailOverMode; /* true se o failover tiver ocorrido */
   dsChar_t        replServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor de replic */
   dsChar_t        homeServerName[DSM_MAX_SERVERNAME_LENGTH+1]; /* nome do servidor inicial */
} tsmInitExOut_t;

#define tsmInitExOutVersion 3

/*------------------------------------------------------------------------+
 |  Definição de tipo para dsmLogExIn_t                                   !
 +------------------------------------------------------------------------*/
typedef struct tsmLogExIn_t
{
   dsUint16_t         stVersion;                  /* versão da estrutura*/
   dsmLogSeverity     severity;
   dsChar_t           appMsgID[8];
   dsmLogType         logType;    /* tipo de log: local, servidor, ambos */
   dsChar_t           *message;   /* texto da mensagem a ser registrado  */
   dsChar_t           appName[DSM_MAX_PLATFORM_LENGTH];
   dsChar_t           osPlatform[DSM_MAX_PLATFORM_LENGTH];
   dsChar_t           appVersion[DSM_MAX_PLATFORM_LENGTH];

} tsmLogExIn_t;

#define tsmLogExInVersion 2

/*------------------------------------------------------------------------+
 |  Definição de tipo para dsmlogExOut_t                                  |
 +------------------------------------------------------------------------*/
typedef struct tsmLogExOut_t
{
   dsUint16_t         stVersion;                  /* versão da estrutura*/
} tsmLogExOut_t;

#define tsmLogExOutVersion 1

/*------------------------------------------------------------------------+
 |  Definição de tipo para dsmRenameIn_t                                  |
 +------------------------------------------------------------------------*/
typedef struct tsmRenameIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsUint32_t      tsmHandle;        /* identificador para a sessão      */
   dsUint8_t      repository;             /* backup ou archive           */
   tsmObjName       *objNameP ;               /* nome do objeto          */
```

```
   dsChar_t       newHl[DSM_MAX_HL_LENGTH + 1] ; /* novo nome de nível superior */
   dsChar_t       newLl[DSM_MAX_LL_LENGTH + 1];  /* novo nome de Nível Inferior */
   dsmBool_t      merge;                          /* associar ao nome existente*/
   ObjID          objId;                  /* objId para Archive          */
} tsmRenameIn_t;

#define tsmRenameInVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para dsmRenameOut_t
 +-------------------------------------------------------------------------*/
typedef struct tsmRenameOut_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
} tsmRenameOut_t;

#define tsmRenameOutVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmEndSendObjExIn_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndSendObjExIn_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
   dsUint32_t      tsmHandle;        /* identificador para a sessão      */
} tsmEndSendObjExIn_t;

#define tsmEndSendObjExInVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para dsmEndSendObjExOut_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndSendObjExOut_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsStruct64_t       totalBytesSent;     /* total de bytes lidos do apl. */
   dsmBool_t          objCompressed;     /* foi compactado por objeto     */
   dsStruct64_t       totalCompressSize; /* tamanho total após compactação */
   dsStruct64_t       totalLFBytesSent; /* total de bytes enviados sem Lan */
   dsUint8_t          encryptionType;  /* tipo de criptografia utilizado  */
   dsmBool_t          objDeduplicated;     /* foi o objeto processado para dedup de dados dist. */
   dsStruct64_t       totalDedupSize;      /* tamanho total após dedup */
}tsmEndSendObjExOut_t;

#define tsmEndSendObjExOutVersion 3

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmGroupHandlerIn_t
 +-------------------------------------------------------------------------*/
typedef struct tsmGroupHandlerIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsUint32_t      tsmHandle;        /* identificador para a sessão      */
   dsUint8_t       groupType;        /* tipo de grupo                    */
   dsUint8_t       actionType;       /* tipo de operação do grupo        */
   dsUint8_t       memberType;       /* tipo de membro: Líder ou membro   */
   dsStruct64_t    leaderObjId;      /* OBJID do líder do grupo          */
   dsChar_t        *uniqueGroupTagP; /* identificador exclusivo do grupo */
   tsmObjName      *objNameP ;       /* nome do objeto do líder do grupo */
   dsmGetList      memberObjList;    /* lista de objetos a remover, designar */
} tsmGroupHandlerIn_t;

#define tsmGroupHandlerInVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmGroupHandlerExOut_t
 +-------------------------------------------------------------------------*/
```

```
typedef struct tsmGroupHandlerOut_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
} tsmGroupHandlerOut_t;

#define tsmGroupHandlerOutVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmEndTxnExIn_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndTxnExIn_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint32_t      tsmHandle;        /* identificador para a sessão      */
   dsUint8_t        vote;
} tsmEndTxnExIn_t;

#define tsmEndTxnExInVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmEndTxnExOut_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndTxnExOut_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint16_t         reason;           /* código de razão                */
   dsStruct64_t       groupLeaderObjId; /* ID do obj. groupLeader no qual foi retornado */
   /* DSM_ACTION_OPEN                */
    dsUint8_t      reserved1;         /* utilização futura              */
    dsUint16_t     reserved2;         /* utilização futura              */
} tsmEndTxnExOut_t;

#define tsmEndTxnExOutVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmEndGetDataExIn_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndGetDataExIn_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint32_t      tsmHandle;        /* identificador para a sessão      */
}tsmEndGetDataExIn_t;

#define tsmEndGetDataExInVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para tsmEndGetDataExOut_t
 +-------------------------------------------------------------------------*/
typedef struct tsmEndGetDataExOut_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint16_t         reason;           /* código de razão                */
   dsStruct64_t   totalLFBytesRecv;  /* total de bytes sem lan recebidos */
}tsmEndGetDataExOut_t;

#define tsmEndGetDataExOutVersion 1

/*-------------------------------------------------------------------------+
 |  Definição de tipo para em tsmRetentionEvent()                          |
 +-------------------------------------------------------------------------*/
typedef struct tsmRetentionEventIn_t
{
   dsUint16_t         stVersion;                    /* versão da estrutura*/
   dsUint32_t         tsmHandle;               /* identificador da sessão */
   dsmEventType_t     eventType;                    /* tipo de evento     */
   dsmObjList_t       objList;                      /* ID do objeto       */
}tsmRetentionEventIn_t;
```

```
#define tsmRetentionEventInVersion 1

/*-----------------------------------------------------------------------+
 |  Definição de tipo para em tsmRetentionEvent()                        |
 +-----------------------------------------------------------------------*/
typedef struct tsmRetentionEventOut_t
{
   dsUint16_t     stVersion;        /* versão da estrutura               */
}tsmRetentionEventOut_t;

#define tsmRetentionEventOutVersion 1

/*-----------------------------------------------------------------------+
 |  Definição de tipo para tsmUpdateObjExIn_t
 +-----------------------------------------------------------------------*/
typedef struct tsmUpdateObjExIn_t
{
   dsUint16_t         stVersion;        /* versão da estrutura            */
   dsUint32_t         tsmHandle;              /* identificador da sessão */
   tsmSendType     sendType;             /* enviar tipo back/arch */
   dsChar_t        *descrP;              /* descrição de archive */
   tsmObjName      *objNameP;            /* objName             */
   tsmObjAttr      *objAttrPtr;          /* atributo           */
   dsUint32_t      objUpdAct;            /* ação de atualização       */
   ObjID           archObjId;            /* objId para archive   */
}tsmUpdateObjExIn_t;

#define tsmUpdateObjExInVersion 1

/*-----------------------------------------------------------------------+
 |  Definição de tipo para tsmUpdateObjExOut_t
 +-----------------------------------------------------------------------*/
typedef struct tsmUpdateObjExOut_t
{
   dsUint16_t         stVersion;                   /* versão da estrutura*/
}tsmUpdateObjExOut_t;

#define tsmUpdateObjExOutVersion 1

#if _OPSYS_TYPE == DS_WINNT
#pragma pack()
#endif


#ifdef _MAC
#pragma options align = reset
#endif
#endif /* _H_TSMAPITD */
/*************************************************************************
* Tivoli Storage Manager                                                 *
* Componente do Cliente da API                                           *
*                                                                        *
* (C) Copyright IBM Corporation 1993,2010                                *
*************************************************************************/

/*************************************************************************
* Nome do Arquivo de Cabeçalho: dsmapips.h
*
* Ambiente:        ******************************************
*                  ** Este é um arquivo de origem específico  **
*                  ** de plataforma                           **
*                  ** com versão para o Windows NT            **
*
*                   ******************************************
*
* Notas de Design:     Este arquivo inclui definições dependentes de
```

```
* plataforma
*
* Nome descritivo: Definições para typedefs e LINKAGE do Tivoli Storage
* Manager
*-----------------------------------------------------------------------*/

#ifndef _H_DSMAPIPS
#define _H_DSMAPIPS


#ifndef _WIN64
#pragma pack(1)
#endif

/*<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>*/
/*                    D E F I N I Ç Õ E S  D E  T I P O S              */
/*<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>*/

/* novo arquivo typedef para a Versão 3   */


#if !defined(DSMAPILIB) || defined (XOPEN_BUILD)

/* suporte para linkage */
#include <windows.h>
#define DSMLINKAGE WINAPI


#define DS_WINNT    22
#define _OPSYS_TYPE DS_WINNT

   typedef signed   char  dsInt8_t;
   typedef unsigned char  dsUint8_t;
   typedef signed   short dsInt16_t;
   typedef unsigned short dsUint16_t;
   typedef signed   long  dsInt32_t;
   typedef unsigned long  dsUint32_t;

/*=== Tipos de caracteres e cadeias ===*/
#ifdef UNICODE
  typedef wchar_t dsChar_t;
  #define dsTEXT(x)       L##x
#else
  typedef char dsChar_t;
  #define dsTEXT(x)       x
#endif  /* !UNICODE */

/*=== typedefs e definições comuns derivados de dsChar_t ===*/
typedef dsChar_t            *dsString_t;

  /* incluído na ordem de restauração estendida */
   typedef struct
   {
     dsUint32_t top;
     dsUint32_t hi_hi;
     dsUint32_t hi_lo;
     dsUint32_t lo_hi;
     dsUint32_t lo_lo;
   } dsUint160_t ;

#if defined(_LONG_LONG)
   typedef __int64              dsInt64_t;
   typedef unsigned __int64     dsUint64_t;
   /*=== Um inteiro "true" de 64 bits não assinado ===*/
   typedef __int64              dsLongLong_t;
#else
typedef struct tagUINT64_t
```

```
   {
      dsUint32_t hi;        /* 32 bits mais significativos. */
      dsUint32_t lo;        /* 32 bits menos significativos. */
   } dsUint64_t;
#endif

/*-------------------------------------------------------------------------+
| Tipo de definição para bool_t                                            |
+-------------------------------------------------------------------------*/
/*
 *  Foi necessário criar um tipo booleano que não fosse incompatível com nenhuma outra
 *  versão predefinida dos sistemas operacionais ou dos sistemas de janelas.
 */
typedef enum
{
   dsmFalse = 0x00,
   dsmTrue  = 0x01
}dsmBool_t ;


/*===  para retrocompatibilidade  ===*/
#define uint8  dsUint8_t
#define int8   dsInt8_t
#define uint16 dsUint16_t
#define int16  dsInt16_t
#define uint32 dsUint32_t
#define int32  dsInt32_t
#define uint64 dsStruct64_t
#define bool_t dsBool_t
#define dsBool_t dsmBool_t
#define bTrue    dsmTrue
#define bFalse   dsmFalse

typedef struct
{
   dsUint32_t hi;        /* 32 bits mais significativos. */
   dsUint32_t lo;        /* 32 bits menos significativos. */
}dsStruct64_t ;

#endif /* DSMAPILIB */



#ifndef _WIN64
#pragma pack()
#endif
#endif  /* _H_DSMAPIPS */
/*************************************************************************
* Tivoli Storage Manager                                                 *
* Componente da Origem Comum                                             *
*                                                                        *
* (C) Copyright IBM Corporation 1993,2016                                *
*************************************************************************/

/*************************************************************************
* Nome do Arquivo de Cabeçalho: release.h
*
* Ambiente:        ***********************************************
*                  ** Este é um arquivo de origem independente   **
*                  ** de plataforma                              **
*                  ***********************************************
*
* Notas de Design: Este arquivo contém as informações comuns sobre
*                  versão.release.nível.subnível real
*
* Nome descritivo: Definições para a versão do Tivoli Storage Manager
*
```

```
* Nota: Este arquivo não deve conter nenhuma informações de LOG ou CMVC. Ele é
enviado com o código da API.
*
*--------------------------------------------------------------------------*/

#ifndef _H_RELEASE
#define _H_RELEASE

#define COMMON_VERSION         8
#define COMMON_RELEASE         1
#define COMMON_LEVEL           0
#define COMMON_SUBLEVEL        0
#define COMMON_DRIVER  dsTEXT("")

#define COMMON_VERSIONTXT "8.1.0.0"

#define SHIPYEARTXT  "2016"
#define SHIPYEARTXTW dsTEXT("2016")
#define TSMPRODTXT  "IBM Tivoli Storage Manager"

/*====================================================================
   As definições de cadeia a seguir são utilizadas para informações de
   VERSION
   e não devem ser convertidas para dsTEXT ou osTEXT.  Elas são usadas
   somente no tempo de link.

   Elas também são utilizadas quando o arquivo Jar é construído em Unix.  Consulte o
   script perl tools/unx/mzbuild/createReleaseJava
====================================================================*/
#define COMMON_VERSION_STR    "8"
#define COMMON_RELEASE_STR    "1"
#define COMMON_LEVEL_STR      "0"
#define COMMON_SUBLEVEL_STR   "0"
#define COMMON_DRIVER_STR     ""

/*=== definições de nomes do produto ===*/
#define COMMON_NAME_DFDSM      1
#define COMMON_NAME_ADSM       2
#define COMMON_NAME_TSM        3
#define COMMON_NAME_ITSM       4
#define COMMON_NAME            COMMON_NAME_ITSM

/*====================================================================
   Versão interna, release e versão do nível (build).  Isso
   deve ser exclusivo para cada versão+release+ptf de um produto.
   Essas informações são registradas nos atributos de arquivos e no
   fluxo de dados para finalidades de diagnóstico.
   NOTA: NÃO MODIFIQUE ESSES VALORES. É POSSÍVEL APENAS INCLUIR NOVAS ENTRADAS!
====================================================================*/
#define COMMON_BUILD_TSM_510   1
#define COMMON_BUILD_TSM_511   2
#define COMMON_BUILD_TSM_515   3
#define COMMON_BUILD_TSM_516   4
#define COMMON_BUILD_TSM_520   5
#define COMMON_BUILD_TSM_522   6
#define COMMON_BUILD_TSM_517   7
#define COMMON_BUILD_TSM_523   8
#define COMMON_BUILD_TSM_530   9
#define COMMON_BUILD_TSM_524   10
#define COMMON_BUILD_TSM_532   11
#define COMMON_BUILD_TSM_533   12
#define COMMON_BUILD_TSM_525   13
#define COMMON_BUILD_TSM_534   14
#define COMMON_BUILD_TSM_540   15
#define COMMON_BUILD_TSM_535   16
#define COMMON_BUILD_TSM_541   17
#define COMMON_BUILD_TSM_550   18
```

```
#define COMMON_BUILD_TSM_542   19
#define COMMON_BUILD_TSM_551   20
#define COMMON_BUILD_TSM_610   21
#define COMMON_BUILD_TSM_552   22
#define COMMON_BUILD_TSM_611   23
#define COMMON_BUILD_TSM_543   24
#define COMMON_BUILD_TSM_620   25
#define COMMON_BUILD_TSM_612   26
#define COMMON_BUILD_TSM_553   27
#define COMMON_BUILD_TSM_613   28
#define COMMON_BUILD_TSM_621   29
#define COMMON_BUILD_TSM_622   30
#define COMMON_BUILD_TSM_614   31
#define COMMON_BUILD_TSM_623   32
#define COMMON_BUILD_TSM_630   33
#define COMMON_BUILD_TSM_615   34
#define COMMON_BUILD_TSM_624   35
#define COMMON_BUILD_TSM_631   36
#define COMMON_BUILD_TSM_640   37
#define COMMON_BUILD_TSM_710   38
#define COMMON_BUILD_TSM_625   39
#define COMMON_BUILD_TSM_641   40
#define COMMON_BUILD_TSM_711   41
#define COMMON_BUILD_TSM_712   42
#define COMMON_BUILD_TSM_713   43
#define COMMON_BUILD_TSM_714   44
#define COMMON_BUILD_TSM_720   45
#define COMMON_BUILD_TSM_721   46
#define COMMON_BUILD_TSM_642   47
#define COMMON_BUILD_TSM_643   48
#define COMMON_BUILD_TSM_715   49
#define COMMON_BUILD_TSM_716   50
#define COMMON_BUILD_TSM_810   51
#define COMMON_BUILD           COMMON_BUILD_TSM_810

/*=== definir VRL como Int para comparações de versão de bitmap ====*/
static const  int  VRL_712 = 712;
static const  int  VRL_713 = 713;
static const  int  VRL_714 = 714;
static const  int  VRL_715 = 715;
static const  int  VRL_716 = 716;
static const  int  VRL_810 = 810;

#define   TDP4VE_PLATFORM_STRING_MBCS     "TDP VMware"
#define   TDP4VE_PLATFORM_STRING          dsTEXT("TDP VMware")

#define   TDP4HYPERV_PLATFORM_STRING_MBCS "TDP HyperV"
#define   TDP4HYPERV_PLATFORM_STRING      dsTEXT("TDP HyperV")

#endif /* _H_RELEASE */
```

Apêndice C. Arquivo de Origem de Definições de Função da API

Este apêndice contém o arquivo de cabeçalho `dsmapifp.h`, portanto, é possível consultar as definições de função para a API.

Nota: DSMLINKAGE é definido de forma diferente para cada sistema operacional. Consulte as definições no arquivo `dsmapips.h` para seu sistema operacional específico.

As informações fornecidas aqui contêm uma cópia point-in-time dos arquivos que são distribuídos com a API. Visualize os arquivos no pacote de distribuição da API para a versão mais recente.

```
/*****************************************************************
* Tivoli Storage Manager                                         *
* Componente do Cliente da API                                   *
*                                                                *
* (C) Copyright IBM Corporation 1993,2002                        *
*****************************************************************/

/*****************************************************************************/
/* Nome do Arquivo de Cabeçalho: dsmapifp.h                                  */
/*                                                                           */
/* Nome descritivo: Protótipos de função da API do Tivoli Storage Manager */
/*****************************************************************************/
#ifndef _H_DSMAPIFP
#define _H_DSMAPIFP

#if defined(__cplusplus)
extern "C" {
#endif

#ifdef DYNALOAD_DSMAPI

/* a função será carregada dinamicamente */
#include "dsmapidl.h"

#else

/* as funções serão carregadas implicitamente a partir da biblioteca */

/*===========================================================================*/
/*              F U N Ç Õ E S  P Ú B L I C A S                               */
/*===========================================================================*/


extern dsInt16_t DSMLINKAGE  dsmBeginGetData(
       dsUint32_t             dsmHandle,
       dsBool_t              mountWait,
       dsmGetType            getType,
       dsmGetList            *dsmGetObjListP
);

extern dsInt16_t DSMLINKAGE  dsmBeginQuery(
       dsUint32_t             dsmHandle,
       dsmQueryType          queryType,
       dsmQueryBuff          *queryBuffer
);

extern dsInt16_t DSMLINKAGE  dsmBeginTxn(
```

```
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE  dsmBindMC(
       dsUint32_t             dsmHandle,
       dsmObjName             *objNameP,
       dsmSendType            sendType,
       mcBindKey             *mcBindKeyP
);

extern dsInt16_t DSMLINKAGE  dsmChangePW(
       dsUint32_t             dsmHandle,
       char                  *oldPW,
       char                  *newPW
);

extern dsInt16_t DSMLINKAGE dsmCleanUp(
       dsBool_t               mtFlag
);


extern dsInt16_t DSMLINKAGE  dsmDeleteAccess(
       dsUint32_t             dsmHandle,
       dsUint32_t            ruleNum
);

extern dsInt16_t DSMLINKAGE  dsmDeleteObj(
       dsUint32_t             dsmHandle,
       dsmDelType            delType,
       dsmDelInfo            delInfo
);

extern dsInt16_t DSMLINKAGE  dsmDeleteFS(
       dsUint32_t             dsmHandle,
       char                  *fsName,
       dsUint8_t              repository
);

extern dsInt16_t DSMLINKAGE  dsmEndGetData(
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE  dsmEndGetDataEx(
       dsmEndGetDataExIn_t  *dsmEndGetDataExInP,
       dsmEndGetDataExOut_t *dsmEndGetDataExOutP
);

extern dsInt16_t DSMLINKAGE  dsmEndGetObj(
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE  dsmEndQuery(
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE  dsmEndSendObj(
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE  dsmEndSendObjEx(
       dsmEndSendObjExIn_t   *dsmEndSendObjExInP,
       dsmEndSendObjExOut_t  *dsmEndSendObjExOutP
);

extern dsInt16_t DSMLINKAGE  dsmEndTxnEx(
       dsmEndTxnExIn_t      *dsmEndTxnExInP,
       dsmEndTxnExOut_t     *dsmEndTxnExOutP
```

```
);

extern dsInt16_t DSMLINKAGE  dsmEndTxn(
       dsUint32_t             dsmHandle,
       dsUint8_t             vote,
       dsUint16_t            *reason
);

extern dsInt16_t DSMLINKAGE  dsmGetData(
       dsUint32_t             dsmHandle,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  dsmGetBufferData(
       getBufferDataIn_t     *dsmGetBufferDataInP,
       getBufferDataOut_t    *dsmGetBufferDataOutP
);

extern dsInt16_t DSMLINKAGE  dsmGetNextQObj(
       dsUint32_t             dsmHandle,
       DataBlk                *dataBlkPtr
) ;

extern dsInt16_t DSMLINKAGE  dsmGetObj(
       dsUint32_t             dsmHandle,
       ObjID                 *objIdP,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  dsmGroupHandler(
       dsmGroupHandlerIn_t   *dsmGroupHandlerInP,
       dsmGroupHandlerOut_t  *dsmGroupHandlerOutP
);

extern dsInt16_t DSMLINKAGE  dsmInit(
       dsUint32_t            *dsmHandle,
       dsmApiVersion         *dsmApiVersionP,
       char                  *clientNodeNameP,
       char                  *clientOwnerNameP,
       char                  *clientPasswordP,
       char                  *applicationType,
       char                  *configfile,
       char                  *options
);

extern dsInt16_t DSMLINKAGE  dsmInitEx(
       dsUint32_t            *dsmHandleP,
       dsmInitExIn_t         *dsmInitExInP,
       dsmInitExOut_t        *dsmInitExOutP
);


extern dsInt16_t DSMLINKAGE  dsmLogEvent(
       dsUint32_t             dsmHandle,
       logInfo               *lopInfoP
);

extern dsInt16_t DSMLINKAGE  dsmLogEventEx(
       dsUint32_t             dsmHandle,
       dsmLogExIn_t          *dsmLogExInP,
       dsmLogExOut_t         *dsmLogExOutP
);

extern dsInt16_t DSMLINKAGE  dsmQueryAccess(
       dsUint32_t             dsmHandle,
       qryRespAccessData     **accessListP,
       dsUint16_t            *numberOfRules
```

```
);

extern void DSMLINKAGE       dsmQueryApiVersion(
       dsmApiVersion         *apiVersionP
);

extern void DSMLINKAGE       dsmQueryApiVersionEx(
       dsmApiVersionEx       *apiVersionP
);

extern dsInt16_t DSMLINKAGE  dsmQueryCliOptions(
       optStruct             *optstructP
);

extern dsInt16_t DSMLINKAGE  dsmQuerySessInfo(
       dsUint32_t             dsmHandle,
       ApiSessInfo           *SessInfoP
);

extern dsInt16_t DSMLINKAGE  dsmQuerySessOptions(
       dsUint32_t             dsmHandle,
       optStruct             *optstructP
);

extern dsInt16_t DSMLINKAGE  dsmRCMsg(
       dsUint32_t             dsmHandle,
       dsInt16_t             dsmRC,
       char                  *msg
);

extern dsInt16_t DSMLINKAGE  dsmRegisterFS(
       dsUint32_t             dsmHandle,
       regFSData             *regFilespaceP
);

extern dsInt16_t DSMLINKAGE   dsmReleaseBuffer(
       releaseBufferIn_t      *dsmReleaseBufferInP,
       releaseBufferOut_t     *dsmReleaseBufferOutP
);

extern dsInt16_t DSMLINKAGE   dsmRenameObj(
       dsmRenameIn_t          *dsmRenameInP,
       dsmRenameOut_t         *dsmRenameOutP
);

extern dsInt16_t DSMLINKAGE  dsmRequestBuffer(
       requestBufferIn_t     *dsmRequestBufferInP,
       requestBufferOut_t    *dsmRequestBufferOutP
);

extern dsInt16_t DSMLINKAGE   dsmRetentionEvent(
       dsmRetentionEventIn_t  *dsmRetentionEventInP,
       dsmRetentionEventOut_t *dsmRetentionEventOutP
);

extern dsInt16_t DSMLINKAGE   dsmSendBufferData(
       sendBufferDataIn_t     *dsmSendBufferDataInP,
       sendBufferDataOut_t    *dsmSendBufferDataOutP
);

extern dsInt16_t DSMLINKAGE   dsmSendData(
       dsUint32_t             dsmHandle,
       DataBlk                *dataBlkPtr
) ;

extern dsInt16_t DSMLINKAGE   dsmSendObj(
       dsUint32_t             dsmHandle,
```

```
       dsmSendType            sendType,
       void                  *sendBuff,
       dsmObjName             *objNameP,
       ObjAttr                *objAttrPtr,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE   dsmSetAccess(
       dsUint32_t             dsmHandle,
       dsmAccessType          accessType,
       dsmObjName             *objNameP,
       char                   *node,
       char                   *owner
);

extern dsInt16_t DSMLINKAGE   dsmSetUp(
       dsBool_t              mtFlag,
       envSetUp               *envSetUpP
);

extern dsInt16_t DSMLINKAGE   dsmTerminate(
       dsUint32_t             dsmHandle
);

extern dsInt16_t DSMLINKAGE   dsmUpdateFS(
       dsUint32_t             dsmHandle,
       char                   *fs,
       dsmFSUpd               *fsUpdP,
       dsUint32_t            fsUpdAct
);

extern dsInt16_t DSMLINKAGE   dsmUpdateObj(
       dsUint32_t             dsmHandle,
       dsmSendType            sendType,
       void                  *sendBuff,
       dsmObjName             *objNameP,
       ObjAttr                *objAttrPtr,
       dsUint32_t            objUpdAct

);

extern dsInt16_t DSMLINKAGE   dsmUpdateObjEx(
       dsmUpdateObjExIn_t     *dsmUpdateObjExInP,
       dsmUpdateObjExOut_t    *dsmUpdateObjExOutP
);


#endif /* ifdef DYNALOAD */

#if defined(__cplusplus)
   }
#endif

#endif /* _H_DSMAPIFP */
```

Esta seção contém as definições de função para a API. É uma cópia do arquivo de cabeçalho `tsmapifp.h`.

Nota: DSMLINKAGE é definido de forma diferente para cada sistema operacional. Consulte as definições no arquivo `tsmapips.h` para seu sistema operacional específico.

```
/*************************************************************************
* Tivoli Storage Manager                                                 *
* Componente do Cliente da API                                           *
*                                                  *
```

```
* (C) Copyright IBM Corporation 1993,2002                              *
*********************************************************************/

/*********************************************************************/
/* Nome do Arquivo de Cabeçalho: tsmapifp.h                          */
/*                                                                   */
/* Nome descritivo: Protótipos de função da API do Tivoli Storage Manager */
/*********************************************************************/
#ifndef _H_TSMAPIFP
#define _H_TSMAPIFP

#if defined(__cplusplus)
extern "C" {
#endif

#ifdef DYNALOAD_DSMAPI

/* a função será carregada dinamicamente */
#include "dsmapidl.h"

#else

/* as funções serão carregadas implicitamente a partir da biblioteca */

/*====================================================================*/
/*              F U N Ç Õ E S  P Ú B L I C A S                        */
/*====================================================================*/


typedef void tsmQueryBuff;

extern dsInt16_t DSMLINKAGE  tsmBeginGetData(
       dsUint32_t            tsmHandle,
       dsBool_t              mountWait,
       tsmGetType            getType,
       dsmGetList            *dsmGetObjListP
);

extern dsInt16_t DSMLINKAGE  tsmBeginQuery(
       dsUint32_t            tsmHandle,
       tsmQueryType          queryType,
       tsmQueryBuff          *queryBuffer
);

extern dsInt16_t DSMLINKAGE  tsmBeginTxn(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmBindMC(
       dsUint32_t            tsmHandle,
       tsmObjName            *objNameP,
       tsmSendType           sendType,
       tsmMcBindKey          *mcBindKeyP
);

extern dsInt16_t DSMLINKAGE  tsmChangePW(
       dsUint32_t            tsmHandle,
       dsChar_t              *oldPW,
       dsChar_t              *newPW
);

extern dsInt16_t DSMLINKAGE tsmCleanUp(
       dsBool_t              mtFlag
);


extern dsInt16_t DSMLINKAGE  tsmDeleteAccess(
```

```
       dsUint32_t            tsmHandle,
       dsUint32_t            ruleNum
);

extern dsInt16_t DSMLINKAGE  tsmDeleteObj(
       dsUint32_t            tsmHandle,
       tsmDelType            delType,
       tsmDelInfo            delInfo
);

extern dsInt16_t DSMLINKAGE  tsmDeleteFS(
       dsUint32_t            tsmHandle,
       dsChar_t              *fsName,
       dsUint8_t             repository
);

extern dsInt16_t DSMLINKAGE  tsmEndGetData(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmEndGetDataEx(
       tsmEndGetDataExIn_t      *tsmEndGetDataExInP,
       tsmEndGetDataExOut_t     *tsmEndGetDataExOutP
);

extern dsInt16_t DSMLINKAGE  tsmEndGetObj(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmEndQuery(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmEndSendObj(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmEndSendObjEx(
       tsmEndSendObjExIn_t      *tsmEndSendObjExInP,
       tsmEndSendObjExOut_t     *tsmEndSendObjExOutP
);

extern dsInt16_t DSMLINKAGE  tsmEndTxn(
       dsUint32_t            tsmHandle,
       dsUint8_t             vote,
       dsUint16_t            *reason
);

extern dsInt16_t DSMLINKAGE  tsmEndTxnEx(
       tsmEndTxnExIn_t      *tsmEndTxnExInP,
       tsmEndTxnExOut_t     *tsmEndTxnExOutP
);

extern dsInt16_t DSMLINKAGE  tsmGetData(
       dsUint32_t            tsmHandle,
       DataBlk*dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  tsmGetBufferData(
       getBufferDataIn_t     *tsmGetBufferDataInP,
       getBufferDataOut_t    *tsmGetBufferDataOutP
);

extern dsInt16_t DSMLINKAGE  tsmGetNextQObj(
       dsUint32_t            tsmHandle,
       DataBlk*dataBlkPtr
) ;
```

```
extern dsInt16_t DSMLINKAGE  tsmGetObj(
       dsUint32_t            tsmHandle,
       ObjID                 *objIdP,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  tsmGroupHandler(
       tsmGroupHandlerIn_t   *tsmGroupHandlerInP,
       tsmGroupHandlerOut_t  *tsmGroupHandlerOutP
);

extern dsInt16_t DSMLINKAGE  tsmInitEx(
       dsUint32_t            *tsmHandleP,
       tsmInitExIn_t         *tsmInitExInP,
       tsmInitExOut_t        *tsmInitExOutP
);


extern dsInt16_t DSMLINKAGE  tsmLogEventEx(
       dsUint32_t            tsmHandle,
       tsmLogExIn_t          *tsmLogExInP,
       tsmLogExOut_t         *tsmLogExOutP
);

extern dsInt16_t DSMLINKAGE  tsmQueryAccess(
       dsUint32_t            tsmHandle,
       tsmQryRespAccessData  **accessListP,
       dsUint16_t            *numberOfRules
);


extern void DSMLINKAGE       tsmQueryApiVersionEx(
       tsmApiVersionEx       *apiVersionP
);

extern dsInt16_t DSMLINKAGE  tsmQueryCliOptions(
       tsmOptStruct          *optstructP
);

extern dsInt16_t DSMLINKAGE  tsmQuerySessInfo(
       dsUint32_t            tsmHandle,
       tsmApiSessInfo        *SessInfoP
);

extern dsInt16_t DSMLINKAGE  tsmQuerySessOptions(
       dsUint32_t            tsmHandle,
       tsmOptStruct          *optstructP
);

extern dsInt16_t DSMLINKAGE  tsmRCMsg(
       dsUint32_t            tsmHandle,
       dsInt16_t             tsmRC,
       dsChar_t              *msg
);

extern dsInt16_t DSMLINKAGE  tsmRegisterFS(
       dsUint32_t            tsmHandle,
       tsmRegFSData          *regFilespaceP
);

extern dsInt16_t DSMLINKAGE  tsmReleaseBuffer(
       releaseBufferIn_t     *tsmReleaseBufferInP,
       releaseBufferOut_t    *tsmReleaseBufferOutP
);

extern dsInt16_t DSMLINKAGE  tsmRenameObj(
```

```
       tsmRenameIn_t         *tsmRenameInP,
       tsmRenameOut_t        *tsmRenameOutP
);

extern dsInt16_t DSMLINKAGE  tsmRequestBuffer(
       requestBufferIn_t     *tsmRequestBufferInP,
       requestBufferOut_t    *tsmRequestBufferOutP
);

extern dsInt16_t DSMLINKAGE   tsmRetentionEvent(
       tsmRetentionEventIn_t  *tsmRetentionEventInP,
       tsmRetentionEventOut_t *tsmRetentionEventOutP
);

extern dsInt16_t DSMLINKAGE   tsmSendBufferData(
       sendBufferDataIn_t     *tsmSendBufferDataInP,
       sendBufferDataOut_t    *tsmSendBufferDataOutP
);

extern dsInt16_t DSMLINKAGE  tsmSendData(
       dsUint32_t            tsmHandle,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  tsmSendObj(
       dsUint32_t            tsmHandle,
       tsmSendType           sendType,
       void                  *sendBuff,
       tsmObjName            *objNameP,
       tsmObjAttr            *objAttrPtr,
       DataBlk                *dataBlkPtr
);

extern dsInt16_t DSMLINKAGE  tsmSetAccess(
       dsUint32_t            tsmHandle,
       tsmAccessType         accessType,
       tsmObjName            *objNameP,
       dsChar_t              *node,
       dsChar_t              *owner
);

extern dsInt16_t DSMLINKAGE tsmSetUp(
       dsBool_t             mtFlag,
       tsmEnvSetUp          *envSetUpP
);

extern dsInt16_t DSMLINKAGE  tsmTerminate(
       dsUint32_t            tsmHandle
);

extern dsInt16_t DSMLINKAGE  tsmUpdateFS(
       dsUint32_t            tsmHandle,
       dsChar_t              *fs,
       tsmFSUpd              *fsUpdP,
       dsUint32_t            fsUpdAct
);

extern dsInt16_t DSMLINKAGE  tsmUpdateObj(
       dsUint32_t            tsmHandle,
       tsmSendType           sendType,
       void                  *sendBuff,
       tsmObjName            *objNameP,
       tsmObjAttr            *objAttrPtr,
       dsUint32_t            objUpdAct

);
```

```
extern dsInt16_t DSMLINKAGE tsmUpdateObjEx(
       tsmUpdateObjExIn_t         *tsmUpdateObjExInP,
       tsmUpdateObjExOut_t        *tsmUpdateObjExOutP
);


#endif /* ifdef DYNALOAD */

#if defined(__cplusplus)
   }
#endif

#endif /* _H_TSMAPIFP */
```

Apêndice D. Recursos de Acessibilidade para a Família de Produtos IBM Spectrum Protect

Os recursos de acessibilidade ajudam os usuários que possuem uma deficiência, como mobilidade restrita ou visão limitada, a usar o conteúdo de tecnologia da informação com êxito.

Visão Geral

A família de produtos IBM Spectrum Protect inclui os principais recursos de acessibilidade a seguir:

- Operação apenas do teclado
- Operações que usam um leitor de tela

A família de produtos IBM Spectrum Protect usa o padrão W3C mais recente, WAI-ARIA 1.0 (www.w3.org/TR/wai-aria/), para assegurar conformidade com o US Section 508 (www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards) e Web Content Accessibility Guidelines (WCAG) 2.0 (www.w3.org/TR/WCAG20/). Para aproveitar os recursos de acessibilidade, use a liberação mais recente do seu leitor de tela e o último navegador da web que seja suportado pelo produto.

A documentação do produto no IBM Knowledge Center é ativada para acessibilidade. Os recursos de acessibilidade do IBM Knowledge Center estão descritos na Seção de acessibilidade da ajuda do IBM Knowledge Center (www.ibm.com/support/knowledgecenter/about/releasenotes.html?view=kc#accessibility).

Navegação pelo Teclado

Esse produto usa as chaves de navegação padrão

Informações sobre a Interface

As interfaces com o usuário não têm conteúdo que pisca 2-55 vezes por segundo.

Interfaces com o usuário da web dependem de folhas de estilo em cascata para renderizar o conteúdo corretamente e para fornecer uma experiência utilizável. O aplicativo fornece uma maneira equivalente para os usuários com visão reduzida usarem as configurações de exibição do sistema, incluindo o modo de alto contraste. É possível controlar o tamanho da fonte usando as configurações do dispositivo ou do navegador da web.

As interfaces com o usuário da web incluem referências de navegação WAI-ARIA que podem ser usadas para navegar rapidamente para áreas funcionais no aplicativo.

Software do Fornecedor

A família de produtos do IBM Spectrum Protect inclui determinado software de fornecedor que não é coberto pelo contrato de licença da IBM. A IBM não representa nenhum recurso de acessibilidade desses produtos. Entre em contato

com o fornecedor para obter informações de acessibilidade sobre estes produtos.

Informações sobre acessibilidade relacionadas

Além dos websites padrão do IBM help desk e do suporte, a IBM tem um serviço telefônico TTY para ser usado por clientes com deficiência auditiva para acessar os serviços de suporte e vendas:

Serviço de TTY
800-IBM-3383 (800-426-3383)
(na América do Norte)

Para obter informações adicionais sobre o compromisso que a IBM tem com a acessibilidade, consulte Acessibilidade IBM(www.ibm.com/able).

Aviso

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. Este material pode estar disponível na IBM em outros idiomas. No entanto, pode ser necessário possuir uma cópia do produto ou da versão de produto no mesmo idioma para acessá-lo.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um produto, programa ou serviço IBM não afirma ou significa que apenas que o produto, programa ou serviço IBM pode ser usado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não concede ao Cliente nenhum direito sobre tais patentes. Pedidos de licenças devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO-INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Esta publicação pode conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode fazer aperfeiçoamentos e/ou alterações nos produtos ou programas descritos nesta publicação a qualquer momento sem aviso prévio.

As referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença de Programa Internacional IBM ou de qualquer outro contrato equivalente entre as partes.

Os dados de desempenho discutidos aqui são apresentados como derivados sob as condições de operação específicas. Os resultados reais podem variar.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas aos fornecedores desses produtos.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. Os programas de amostra são fornecidos "NO ESTADO EM QUE SE ENCONTRAM", sem

garantia de qualquer tipo. A IBM não poderá ser responsabilizada por quaisquer danos decorrentes ao uso dos programas de amostra.

Qualquer cópia, parte desses programas de amostra ou trabalho derivado deve incluir um aviso de copyright da seguinte forma: © (o nome de sua empresa) (ano). Partes deste código são derivadas dos Programas de Amostra da IBM Corp. © Copyright IBM Corp. _insira o ano ou anos_.

Marcas

IBM, o logotipo IBM e ibm.com são marcas registradas ou comerciais da International Business Machines Corp., registradas em vários países no mundo todo. Outros nomes de produtos e serviços podem ser marcas registradas da IBM ou de outras empresas. Uma lista atual de marcas comerciais IBM está disponível na web em "Copyright and trademark information" em www.ibm.com/legal/copytrade.shtml.

Adobe é uma marca registrada da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

Linear Tape-Open, LTO e Ultrium são marcas comerciais da HP, IBM Corp. e Quantum nos Estados Unidos e em outros países.

Intel e Itanium são marcas comerciais ou marcas registradas da Intel Corporation ou de suas subsidiárias nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows e Windows NT são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Java™ e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.

SoftLayer é uma marca registrada da SoftLayer, Inc., uma empresa IBM.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Termos e Condições para a Documentação do Produto

As permissões para uso dessas publicações são concedidas sujeitas aos termos e condições a seguir.

Aplicabilidade

Esses termos e condições são adicionais a quaisquer termos de uso para o website da IBM.

utilizar o Personal

Você pode reproduzir estas publicações para seu uso pessoal não comercial desde que todos os avisos do proprietário sejam preservados. O Cliente não pode distribuir, exibir ou fazer trabalho derivado destas publicações, ou de parte delas, sem o consentimento expresso da IBM.

Uso comercial

É possível reproduzir, distribuir e exibir estas publicações exclusivamente

dentro de sua empresa desde que todos os avisos do proprietário sejam preservados. O Cliente não pode fazer trabalhos derivados destas publicações ou reproduzir, distribuir ou exibir estas publicações, ou qualquer parte delas, fora de sua empresa, sem o consentimento expresso da IBM.

Direitos

Exceto como expressamente concedido nesta permissão, nenhuma outra permissão, licença ou direito é concedido, seja expresso ou implícito, para as publicações ou para quaisquer informações, dados, software ou outra propriedade intelectual nelas contidos.

A IBM reserva-se o direito de retirar as permissões concedidas aqui sempre que, a seu critério, o uso das publicações prejudicar seus interesses ou, conforme determinação da IBM, as instruções anteriores não estão sendo seguidas adequadamente.

O Cliente não pode fazer download, exportar ou reexportar estas informações, exceto em conformidade total com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação dos Estados Unidos.

A IBM NÃO GARANTE O CONTEÚDO DESTAS PUBLICAÇÕES. AS PUBLICAÇÕES SÃO FORNECIDAS "NO ESTADO EM QUE SE ENCONTRAM", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO A, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, NÃO INFRAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Considerações sobre política de privacidade

Os produtos de Software IBM, incluindo as soluções de software como serviço (“Ofertas de Software”), podem usar cookies ou outras tecnologias para coletar informações sobre o uso do produto, para ajudar a melhorar a experiência do usuário final, para customizar interações com o usuário final ou para outros propósitos. Em muitos casos, nenhuma informação pessoalmente identificável é coletada pelas Ofertas de Software. Algumas de nossas Ofertas de Software podem permitir a coleta de informações identificáveis pessoalmente. Se esta Oferta de Software usar cookies para coletar informações de identificação pessoal, informações específicas sobre o uso de cookies desta oferta serão apresentadas abaixo.

Esta Oferta de Software não usa cookies ou outras tecnologias para coletar informações pessoalmente identificáveis.

Se as configurações implementadas para esta Oferta de software fornecerem a você, como cliente, a capacidade de coletar informações de identificação pessoal de usuários finais por meio de cookies e outras tecnologias, é necessário buscar seu próprio conselho jurídico legal sobre quaisquer leis aplicáveis a este tipo de coleção de dados, incluindo quaisquer requisitos de aviso e consentimento.

Para obter informações adicionais sobre o uso de várias tecnologias, incluindo cookies, para estes propósitos, consulte a Política de privacidade da IBM em http://www.ibm.com/privacy e a Declaração de privacidade on-line da IBM em http://www.ibm.com/privacy/details na seção intitulada “Cookies, Web Beacons and Other Technologies” e “IBM Software Products and Software-as-a-Service Privacy Statement” em http://www.ibm.com/software/info/product-privacy.

Glossário

Está disponível um glossário com termos e definições para a família de produtos IBM Spectrum Protect.

Consulte o IBM Spectrum Protectglossário.

Para visualizar glossários para outros produtos IBM, consulte IBM Terminology.

Índice Remissivo

Numéricos

A

B

C

D

E

F

G

I

T

U

V

W

Número do Programa: 5725-W98
5725-W99
5725-X15

Impresso no Brasil