

AIX Versão 7.1

Cluster Data Aggregation Tool
Guia de Usuário e de Referência

IBM

AIX Versão 7.1

Cluster Data Aggregation Tool
Guia de Usuário e de Referência

IBM

Nota

Antes de utilizar estas informações e o produto suportado por elas, leia as informações no “Avisos” na página 23.

Esta edição se aplica ao AIX Versão 7.1 e a todas as liberações e modificações subsequentes até que indicado de outra forma em novas edições.

© Copyright IBM Corporation 2010, 2014.

Índice

Sobre Esse Documento	v	Tipos de Coleta	16
Realçando	v	Tipos de Coleta Padrão	16
Distinção Entre Maiúsculas e Minúsculas	v	Estendendo a Estrutura	17
ISO 9000.	v	Scripts de Coleta Customizados	18
		Auxiliares de Estrutura	20
		Iniciação Rápida CDAT	21
Guia de Usuário e de Referência de CDAT	1	Avisos	23
Cluster Data Aggregation Tool	1	Considerações sobre política de privacidade	25
Visão Geral	1	Marcas comerciais	25
Comando CDAT	2		
Coletas Periódicas	15	Índice Remissivo	27
Arquivos de Log	15		

Sobre Esse Documento

Este documento fornece a usuários e administradores de sistema informações completas sobre a Cluster Data Aggregation Tool (CDAT). A AIX Cluster Data Aggregation Tool fornece uma única instância para ativar a depuração de RAS e ações de monitoramento e para coletar dados de determinação de problema para diversos nós.

Realçando

As seguintes convenções de destaque são usadas neste documento:

Negrito	Identifica comandos, sub-rotinas, palavras-chaves, arquivos, estruturas, diretórios e outros itens cujos nomes são predefinidos pelo sistema. Identifica também objetos gráficos, como botões, rótulos e ícones que o usuário seleciona.
<i>Itálico</i>	Identifica parâmetros cujos nomes ou valores reais devem ser fornecidos pelo usuário.
Monoespaçamento	Identifica exemplos de valores de dados específicos, exemplos de texto semelhante ao que você pode visualizar, exemplos de partes de código de programa semelhantes ao que você pode gravar como um programador, mensagens do sistema ou informações que devem realmente ser inseridas.

Distinção Entre Maiúsculas e Minúsculas

Tudo no sistema operacional AIX faz distinção entre maiúsculas e minúsculas, ou seja, diferencia letras maiúsculas de minúsculas. Por exemplo, você pode usar o comando **ls** para listar arquivos. Se você digitar **LS**, o sistema responderá que o comando não foi localizado. Da mesma forma, **FILEA**, **FiLea** e **filea** são três nomes de arquivo distinto, mesmo se eles residirem no mesmo diretório. Para evitar que ações indesejadas sejam executadas, certifique-se de usar sempre a distinção entre maiúsculas e minúscula correta.

ISO 9000

Os sistemas de qualidade registrados ISO 9000 foram utilizados no desenvolvimento e fabricação deste produto.

Guia de Usuário e de Referência de CDAT

A CDAT (Cluster Data Aggregation Tool) fornece uma única instância para ativar a depuração de RAS e ações de monitoramento e para coletar dados de determinação de problema para diversos nós. Para usar essas informações efetivamente, você deve familiarizar-se com os comandos, chamadas do sistema, sub-rotinas, formatos de arquivo e arquivos especiais. Este tópico também está disponível no CD da documentação enviada com o sistema operacional AIX.

O ambiente da Ferramenta de Cluster Data Aggregation consiste em um nó central principal e em nós remotos. O Cluster Data Aggregation Tool é instalado em e executado a partir do nó principal central. O nó principal central hospeda o repositório de coleta de dados, que é um novo sistema de arquivos que contém todos os dados de coleta a partir de vários nós remotos. Os nós remotos são os locais em que os dados de Cluster Data Aggregation Tool são coletados, que são AIX LPAR, VIOS e HMC. A Ferramenta de Cluster Data Aggregation é gerenciada pelo comando **cdat** que é dividida em diversos subcomandos. Os subcomandos são **access**, **archive**, **check**, **collect**, **delete**, **discover-nodes**, **init**, **list-nodes**, **list-types** e **show**.

Cluster Data Aggregation Tool

A AIX Cluster Data Aggregation Tool fornece uma única instância para ativar a depuração de RAS e ações de monitoramento e para coletar dados de determinação de problema para diversos nós. O ambiente da Ferramenta de Cluster Data Aggregation consiste em um nó central principal e em nós remotos.

Visão Geral

A AIX Cluster Data Aggregation Tool fornece uma única instância para ativar a depuração de RAS e ações de monitoramento e para coletar dados de determinação de problema para diversos nós.

O Cluster Data Aggregation Tool é instalado em e executado a partir do nó principal central. O nó principal central hospeda o repositório de coleta de dados, que é um novo sistema de arquivos que contém todos os dados de coleta a partir de vários nós remotos. Os nós remotos estão onde os dados de Cluster Data Aggregation Tool são coletados, que são AIX LPAR, VIOS e HMC.

O Cluster Data Aggregation Tool possui os seguintes recursos.

- Captura dados de determinação de problemas através de nós múltiplos.
- Suporta o envio de ferramenta de reunião de dados nos nós remotos.
- Fornece um recurso de plug-in extensível que suporta novos tipos de coleta de dados para usuário.
- Integra-se ao RBAC para possibilitar que o usuário não raiz colete dados de Cluster Data Aggregation Tool.
- Confia em SSH para uma conectividade segura entre nós.
- É controlado via painel CLI e SMIT.

O comando Cluster Data Aggregation Tool é chamado **cdat**. É dividido em vários subcomandos. Os subcomandos são **init**, **show**, **check**, **delete**, **discover-nodes**, **list-nodes**, **access**, **collect**, **list-types** e **archive**. O subcomando **init** cria a infraestrutura de dados e define o usuário usado para executar todos os outros subcomandos.

O Cluster Data Aggregation Tool fornece uma interface smit. Ele pode ser ativado a partir do menu Determinação de Problemas ou usando o atalho **cdat** (smit cdat).

Para capturar dados de determinação de problema em diversos nós, a estrutura de coleta fornece os seguintes recursos:

- Ele coordena a chamada da reunião de dados do Cluster Data Aggregation Tool. Para um second failure data capture (SFDC), ele substitui a necessidade dos clientes em iniciar manualmente a reunião de dados entre vários nós e sincroniza o início e a parada da instância de coleta de dados entre vários nós.
- Recuperar o RAS e monitorar dados de vários nós, ele fornece uma maneira fácil de coletá-los a partir de vários nós remotos e colocá-los em um único nó central. As ferramentas do RAS precisam ser distribuídas para vários nós.
- Os dados de estrutura de Cluster Data Aggregation Tool podem ser reunidos a partir do AIX LPAR, VIOS e HMC
- O nó central suporta apenas o AIX. Os nós remotos suportam HMC (Linux) e AIX. O nó central como um host suporta AIX Versão 6.1 e releases superiores.

O repositório de coleta de dados deve ser grande o suficiente para conter todas as coletas de dados em um único local. O tamanho padrão é de 10 GB.

A coleta de tipo de dados do Cluster Data Aggregation Tool suporta coletas de dados de algumas ferramentas, como **perfpmr**, **snap** e **trace**. A estrutura do Cluster Data Aggregation Tool fornece a capacidade para conectar tipos de coleta de dados adicionais conforme necessário.

A estrutura do Cluster Data Aggregation Tool é inicializada pelo usuário raiz. Um usuário não raiz é criado durante a inicialização da ferramenta e é usado por subcomandos de ferramenta subsequentes. Além disso, a estrutura permite criar um usuário não administrador nos nós remotos que serão usados durante a coleta de dados. O usuário recebe funções do AIX com autorizações necessárias que assegurarão que os comandos RAS (**snap**, **dump**, **trace**, **iptrace**, **perfpmr**) possam ser executados por esse usuário.

O núcleo da estrutura do Cluster Data Aggregation Tool é uma base de linha de comandos independente. O usuário raiz ou o usuário não administrador especificado pode inicializar o Cluster Data Aggregation Tool e capturar dados ao executar os comandos.

A configuração do Cluster Data Aggregation Tool é simples. A Estrutura do Cluster Data Aggregation Tool suporta mudanças na configuração, como um usuário que pode desejar incluir ou remover nós ou alterar as informações de configuração para um nó.

Comando CDAT

O comando Cluster Data Aggregation Tool é chamado **cdat**.

Finalidade

A Cluster Data Aggregation Tool é dividida em vários subcomandos.

Descrição

O comando **cdat** é dividido em vários subcomandos. Os subcomandos são **init**, **show**, **check**, **delete**, **discover-nodes**, **list-nodes**, **access**, **collect**, **list-types** e **archive**. Apenas o subcomando **init** precisa ser executado pelo usuário privilegiado (raiz). O subcomando **init** cria a infraestrutura de dados e define o usuário usado para executar todos os outros subcomandos.

Nota: Para evitar acessos simultâneos ao arquivo **nodes.txt** ou ao arquivo **collect.xml**, é proibido executar diversas instâncias do comando **cdat** no mesmo repositório e o repositório é protegido por um arquivo de bloqueio.

Sintaxe

```
cdat -h <subcommand>  
[<options....>]
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
subcomando	Especifica o nome do subcomando, entre eles: <ul style="list-style-type: none">• inicialização• show• check• excluir• discover-nodes• list-nodes• access• coletar• list-types• archive

Saída

Se for usado com **-h** ou com um subcomando desconhecido, o comando imprimirá a ajuda, caso contrário, a saída será a saída do subcomando.

Código de Retorno

Se usado com **-h** ou com um subcomando desconhecido, o código de retorno será **1**. Caso contrário, o código de retorno será o código de retorno do subcomando.

Subcomando **cdat init**

Finalidade

Inicializa o repositório de Cluster Data Aggregation.

Descrição

O subcomando **init** inicializa o repositório do Cluster Data Aggregation. Esse subcomando deve ser executado pelo usuário **raiz** privilegiado.

Você pode especificar o nome do diretório para o repositório (**/cdat** por padrão). É possível solicitar a criação de um sistema de arquivos específico. Nesse caso, você pode especificar o nome do grupo de volumes e o tamanho. Você pode especificar o volume lógico. Ele também permite definir qual usuário executará o comando **cdat** para coletar dados. O usuário padrão é **cdat**.

Sintaxe

```
cdat init -h  
cdat init [-c [-g VGName] [-s FSSize]] [-d Directory] [-l LVName] [-u  
User]
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
-d	Especifica o diretório a ser usado como o repositório.
Diretório	
-u	Especifica o usuário a executar o subcomando cdat .
Usuário	
-c	Cria um volume lógico montado no caminho do diretório cdat .
-g	Seleciona o grupo de volumes a ser usado para criar o volume lógico.
VGName	
-l	Especifica o nome do novo volume lógico a ser usado.
LVName	
-s	Especifica o tamanho do volume lógico.
FSSize	

Saída

Se o subcomando é usado com **-h** ou com um parâmetro inválido, o subcomando imprime a ajuda. O comando exibe o nome do usuário **cdat** e depois solicita uma senha. O comando exibe o nome do diretório que é usado para armazenar os dados de coleta.

Exemplo

```
# cdat init
Creating user "cdat"
Changing password for "cdat"
cdat's new password: *****
Re-enter cdat's new password: *****
creating directory "/cdat"
```

Código de Retorno

Se o subcomando for usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando falhar em criar o usuário ou em configurar a senha do usuário, o código de retorno será 2. Se o comando falhar em criar o diretório, o código de retorno será 3. Se o comando falhar em criar o volume lógico, o código de retorno será 4. Caso contrário, o código de retorno será 0.

Subcomando **cdat show**

Exibe conteúdo do repositório de Cluster Data Aggregation.

Finalidade

Exibe o conteúdo do repositório de Cluster Data Aggregation.

Descrição

O subcomando **show** exibe o conteúdo do repositório de Cluster Data Aggregation. Um primeiro nível de detalhamento apenas exibe informações de coleta globais. Um segundo nível também exibe as informações do nó.

É possível especificar a exibição por nó ou por ID de coleta (por padrão). É possível especificar um ID de coleta, um número de PMR ou um nó para filtrar a saída. É possível usar o modo detalhado para exibir mais informações.

Sintaxe

```
cdat show -h
cdat show [-v]
cdat show [-v] Id
cdat show [-v] -p PMR
cdat show [-v] -n [Host]
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
-v	Ativa o modo detalhado. Exibe informações de nó.
-n	Exibe a lista ordenada por nó.
Id	Especifica o nome da coleta.
-p	Especifica o número de PMR da coleta.
PMR	

Saída

Se o subcomando é usado com **-h** ou com um parâmetro inválido, o subcomando imprime a ajuda. Caso contrário, o comando exibe informações solicitadas.

O comando **cdat show** sem especificar um parâmetro exibe a lista de coletas:

```
# cdat show
Repository: /cdat
Local user: cdat

1: 20090127-12:23:45+0200

    Coletar dados perfpmr para identificar a causa do problema de desempenho
    PMR: 12345,678,901
    Local: /cdat/00000001/

2: 20090212-18:30:25+0200

    Reunir informações de configuração do sistema com snap para análise
    PMR: 12345,589,235
    Local: /cdat/00000002/
```

Com o parâmetro **-v**, a saída é mais detalhada e exibe os nós envolvidos para cada coleta:

```
#
cdat show -v
Repository: /cdat
Local user: cdat

1: 20090127-12:23:45+0200

    Coletar dados perfpmr para identificar a causa do problema de desempenho
    PMR: 12345,678,901
    Local: /cdat/00000001/

    node1:
        type : VIOS
        user : padmin
        machine id: 000069EAD300
        lpar id : 1
        timezone : CEST

    node2:
        type : LPAR
        user : root
        machine id: 000069EAD300
```

```
lpar id : 2
timezone: CEST
```

```
node3:
  type : LPAR
  user : root
  machine id: 000069EAD300
  lpar id : 4
  timezone : CDT
```

```
2: 20090212-18:30:25+0200
```

```
Reunir informações de configuração do sistema com snap para análise.
PMR: 12345,589,235
Local: /cdat/00000002/
```

```
[...]
```

As informações para somente uma coleta podem ser exibidas fornecendo o ID de coleta:

```
# cdat show 1
Repository: /cdat
Local user: cdat
```

```
1: 20090127-12:23:45+0200
```

```
Coletar dados perfpmr para identificar a causa do problema de desempenho.
PMR: 12345,678,901
Local: /cdat/00000001/
```

A opção **-v** também está disponível para exibir as informações dos nós.

A lista pode ser exibida também por nó em vez de por coleta:

```
# cdat show -n
Repository: /cdat
Local user: cdat
```

```
node1:
```

```
1: 20090127-12:23:45+0200
```

```
Coletar dados perfpmr para identificar a causa do problema de desempenho.
PMR: 12345,678,901
Local: /cdat/00000001/
```

```
2: 20090212-18:30:25+0200
```

```
Reunir informações de configuração do sistema com snap para análise.
PMR: 12345,589,235
Local: /cdat/00000002/
```

```
node2:
```

```
1: 20090127-12:23:45+0200
```

```
Coletar dados perfpmr para identificar a causa do problema de desempenho
PMR: 12345,678,901
Local: /cdat/00000001/
```

```
2: 20090127-12:52:07+0200
```

```
Coletar análise de rastreamento de IP.
PMR: 12345,678,901
Local: /cdat/00000002/
```

```
[...]
```

As informações para um determinado nó estão disponíveis usando a opção **-n**:

```
# cdat show -n node1
Repository: /cdat
Local user: cdat

node1:

1: 20090127-12:23:45+0200

    Coletar dados perfpmr para identificar a causa do problema de desempenho.
    PMR: 12345,678,901
    Local: /cdat/00000001/

2: 20090212-18:30:25+0200

    Reunir informações de configuração do sistema com snap para análise.
    PMR: 12345,589,235
    Local: /cdat/00000002/
```

A opção **-v** também está disponível para exibir informações de nós para cada coleta.

A lista de coletas para um determinado PMR está disponível usando a opção **-p**:

```
# cdat show -p 12345,678,901
Repository: /cdat
Local user: cdat

5: 20090127-12:23:45+0200

    Coletar dados perfpmr para identificar a causa do problema de desempenho.
    PMR: 12345,678,901
    Local: /cdat/00000005/

8: 20090212-18:30:25+0200

    Reunir informações de configuração do sistema com snap para análise.
    PMR: 12345,678,901
    Local: /cdat/00000008/
```

A opção **-v** também está disponível para exibir informações de nós para cada coleta.

Código de Retorno

Se o subcomando for usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando falhar, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Subcomando **cdat check**

Para verificar a consistência do repositório de Cluster Data Aggregation, o subcomando **check** é usado.

Finalidade

O subcomando **check** verifica a consistência do repositório do Cluster Data Aggregation.

Descrição

O subcomando **check** verifica a consistência entre o arquivo **cdat.xml** que contém a descrição do repositório e o conteúdo real do repositório de Cluster Data Aggregation.

Se especificar a opção **-d**, ela pode corrigir possíveis inconsistências (esta operação é interativa).

Sintaxe

```
cdat check -h
cdat check [-d]
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
-d	Especifica os arquivos que devem ser corrigidos, se necessário.

Saída

Se for usado com **-h** ou com um parâmetro inválido, o subcomando imprimirá a ajuda. Durante o processamento, o comando exibirá a lista de inconsistências descobertas. Ele pede confirmação antes de reparar.

Código de Retorno

Se usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando detectar alguma inconsistência, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Subcomando **cdat delete**

Finalidade

Remove as coletas especificadas do repositório de Cluster Data Aggregation.

Descrição

O subcomando **delete** remove entradas do arquivo **cdat.xml** e do repositório de Cluster Data Aggregation para as coletas especificadas.

Cada coleta é identificada por um ID. É possível especificar um ID de coleta para suprimir a coleta especificada ou é possível especificar um número de PMR para suprimir todas as coletas relativas ao número de PMR especificado.

Sintaxe

```
cdat delete -h
cdat delete -p PMR
cdat delete Id
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
Id	Especifica o ID da coleta a excluir.
-p	Especifica o número de PMR das coletas a serem excluídas.
PMR	

Saída

Se o subcomando é usado com **-h** ou com um parâmetro inválido, o subcomando imprime a ajuda. Caso contrário, o comando exibe a lista de coletas que está excluindo.

Código de Retorno

Se o subcomando for usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando falhar em excluir a coleta, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Subcomando `cdat discover-nodes`

Finalidade

Recupera o nome da LPAR de todos os nós conectados a um ou mais determinados HMCs ou IVMs.

Descrição

O subcomando `discover-nodes` recupera o nome do LPAR conectado à lista especificada de HMCs. É possível especificar um Servidor de E/S Virtual (VIOS) em vez de um HMC no caso de um Integrated Virtualization Management (IVM) substituir um HMC. Também é possível especificar um nó LPAR para recuperar a partição de carga de trabalho (WPAR) em execução neste LPAR. Você pode especificar o arquivo onde a lista de nós localizados está armazenada. Por padrão, a lista está armazenada no arquivo `nodes.txt` no diretório especificado com o subcomando `cdat init` (`/cdat` por padrão). É possível executar o subcomando `cdat discover-nodes` várias vezes e é possível anexar ou sobrescrever o arquivo. Várias instâncias do mesmo nó não são registradas no arquivo.

O resultado do subcomando `discover-nodes` pode ser usado como entrada aos subcomandos `access` e `run`. O subcomando `discover-nodes` recupera nomes de LPAR, o que significa que se o nome de LPAR não for o mesmo que o nome do host, você deverá editar o arquivo para configurar o nome real do host.

Sintaxe

```
cdat discover-nodes -h
cdat discover-nodes [-a|-w] [-f File] -n Type:[User@]Node ...
```

Sinalizadores

Sinalizador	Descrição
<code>-h</code>	Exibe o uso do comando.
<code>-w</code>	Especifica que o arquivo deve ser sobrescrito.
<code>-a</code>	Especifica que novos nós devem ser anexados ao arquivo.
<code>-f</code>	Especifica o arquivo onde os nós devem ser armazenados.
Arquivo	
<code>-n</code>	Especifica uma lista de nós, em que Type é um dos valores a seguir:
Type: [User@]Node	<ul style="list-style-type: none">• HMC• VIOS• LPAR
	Define o HMC, VIOS ou LPAR ao qual se conectar e possivelmente o usuário usado para conectar.

Saída

Se o subcomando é usado com `-h` ou com um parâmetro inválido, o subcomando imprime a ajuda. Caso contrário, o comando solicita a senha para o usuário `hscroot` (usuário privilegiado em um HMC), o usuário `padmin` (usuário privilegiado em um IVM), o usuário `root` (usuário privilegiado em um LPAR) ou o usuário especificado. O subcomando exibe o nome do arquivo em que a lista de nós está gravada.

Exemplo

```
$ cdat discover-nodes -a -n
HMC:uranus -n LPAR:mylpar
hscroot@uranus's Password: *****
root@mylpar's Password: *****
Updating /cdat/nodes.txt
```

Código de Retorno

Se o subcomando for usado com `-h` ou com um parâmetro inválido, o código de retorno será 1. Se o comando não puder se conectar ao HMC ou IVM, o código de retorno será 2. Se o comando não puder

gravar dados no arquivo, o código de retorno será 3. Caso contrário, o código de retorno será 0.

Subcomando `cdat list-nodes`

Finalidade

Exibe a lista de nós conhecidos.

Descrição

O subcomando `list-nodes` exibe a lista de nós conhecidos (o conteúdo do arquivo `nodes.txt`). Você pode especificar um ou mais arquivos de nós.

Sintaxe

```
cdat list-nodes -h
cdat list-nodes [-f File ...]
```

Sinalizadores

Sinalizador	Descrição
<code>-h</code>	Exibe o uso do comando.
<code>-f</code>	Especifica o arquivo que contém a lista de nós. Vários arquivos podem ser especificados usando várias opções <code>-f</code> .
Arquivo	

Saída

Se o subcomando é usado com `-h` ou com um parâmetro inválido, o subcomando imprime a ajuda. Caso contrário, o comando lista os nós remotos conhecidos.

Exemplo

```
$ cdat list-nodes
HMC uranus
VIOS miranda
LPAR ariel
LPAR umbriel
LPAR titania
LPAR oberon
```

Código de Retorno

Se o subcomando for usado com `-h` ou com um parâmetro inválido, o código de retorno será 1. Se o comando não puder listar os nós, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Subcomando `cdat access`

A autorização de acesso a nós remotos é gerenciada pelo subcomando `access`.

Finalidade

O subcomando `access` gerencia a autorização de acesso aos nós remotos.

Descrição

O subcomando `access` configura a autorização de acesso a nós remotos especificados. Ele cria os usuários especificados nos nós remotos, se eles ainda não existirem, os atributos para esses usuários e todas as autorizações RBAC necessárias para executar a coleta de dados RAS. Ele utiliza o usuário privilegiado apropriado para criar o usuário em cada nó. De acordo, o usuário `hscroot` é usado em HMC, `raiz` em LPAR e `padmin` no VIOS. O subcomando instala a chave pública SSH do usuário `cdat` nos nós remotos. Se o daemon SSH não estiver disponível para um nó remoto, ele usa o protocolo exec (porta 512) se ele

estiver disponível ou usa o protocolo telnet para executar comandos no nó remoto. Você pode especificar diretamente uma lista de nós na linha de comandos ou pode especificar o arquivo que contém a lista de nós. Você pode especificar um usuário remoto padrão que será usado se você não especificar um usuário para um determinado nó. Se você especificar a opção **-d**, o subcomando de acesso **cdat** removerá a autorização de acesso para os nós remotos especificados anteriormente configurados; ele também remove os usuários remotos nos nós remotos se eles foram criados anteriormente pelo subcomando **cdat access** .

Sintaxe

```
cdat access -h
cdat access [-dF] [-u User] -n Type:[User@]Node ...
cdat access [-dF] [-u User] -f File ...
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
-d	Exclui credenciais de autenticação de nós remotos.
-F	Força operação mesmo se já tiver sido feita.
-u	Especifica o usuário a ser criado nos nós remotos.
Usuário	
-n	Especifica a lista de nós com o qual se autenticará, em que Type é um de:
Type:[User@]Node	<ul style="list-style-type: none">• HMC• LPAR• VIOS Node é o nome ou o endereço IP do nó com o qual se conectará.
	User é o usuário a ser criado nesse determinado nó (substitui -u).
-f	Especifica o arquivo que contém a lista de nós com os quais se autenticará. Vários arquivos podem ser especificados usando várias opções -f .
Arquivo	

Saída

Se for usado com **-h** ou com um parâmetro inválido, o subcomando imprimirá a ajuda. Caso contrário, o comando solicitará a senha do usuário **cdat** e exibirá um status da conexão.

Exemplo

```
$ cdat access -u cdat
"cdat" user password: *****
accessing cdat@uranus
accessing cdat@miranda
accessing cdat@ariel
accessing cdat@umbriel
accessing cdat@titania
accessing cdat@oberon
```

Código de Retorno

Se usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando falhar ao se conectar com um nó remoto, o código de retorno será 2. Se o comando falhar ao acessar o arquivo de lista de nós, o código de retorno será 3. Caso contrário o código de retorno será 0.

Subcomando **cdat collect**

Finalidade

Inicia ferramentas de análise em nós remotos e coleta resultados no final.

Descrição

O subcomando **collect** inicia as ferramentas de análise nos nós remotos e coleta os resultados no final.

Da mesma forma para o subcomando **access**, a lista de nós pode ser fornecida na linha de comandos ou a partir do arquivo.

Vários tipos de coleta podem ser feitas em uma coleta. Este subcomando atualiza o arquivo **cdat.xml**, cria o arquivo **collect.xml** e obtém arquivos remotos de nós para posicioná-los no repositório local de Cluster Data Aggregation.

As coletas podem estar associadas a um número de solicitação de modificação de produto (PMR). É mais fácil listar tipos de coleta relacionados ao mesmo PMR (usando o subcomando **show**) ou remover todos os tipos de coleta relacionados a um PMR especificado (usando o subcomando **delete**).

Sintaxe

```
cdat collect [-gqv] [-i Id]
[-p PMR] [-m Comment] [-u User] -t Type[,Options] ... -n Type:[User@]Node
...
cdat collect [-gqv] [-i Id] [-p PMR] [-m
Comment] [-u User] -t Type[,Options] ... -f File ...
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
-i	Especifica o nome da coleta.
Id	
-p	Especifica o número do PMR dessa coleta.
PMR	
-m	Especifica o comentário para essa coleta.
Comentário	
-q	Possibilita modo silencioso (não interativo). Nenhuma pergunta é feita ao usuário. O sinalizador é útil para planejar coletas a partir de uma tarefa cron.
-v	Possibilita o modo detalhado e exibe informações de status adicionais durante a coleta.
-g	Causa o crescimento do sistema de arquivos automaticamente se necessário.
-u	Especifica o usuário a se conectar aos nós remotos.
Usuário	
-t	Especifica o tipo de operação de coleta a executar. Opcionalmente, você pode especificar opções relacionadas ao tipo de coleta desejado.
Type[,Options]	
-n	Especifica os nós aos quais se conectar e, opcionalmente, o usuário a usar.
Type:[User@]Node	
-f	Especifica o arquivo que contém a lista de nós com os quais se conectará. Vários arquivos podem ser especificados usando várias opções -f .
Arquivo	

Saída

Se o subcomando for usado com **-h** ou com um parâmetro inválido, o subcomando imprime a ajuda.

Código de Retorno

Se o subcomando for usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se a operação de coleta falhar, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Informações relacionadas:

“Scripts de Coleta Customizados” na página 18

É possível localizar as informações em um conjunto de seis fases para a operação de coleta: verificar,

iniciar, executar, terminar, capturar e limpar que a estrutura de Cluster Data Aggregation Tool define.

Subcomando `cdat list-types`

Finalidade

Exibe a lista de tipos de coleta instalados e suas descrições.

Descrição

O subcomando `list-types` exibe a lista de tipos de coleta instalados junto com as descrições. Procura por tipos de coleta nos diretórios `/usr/lib/cdat/types/` e `/var/adm/ras/cdat/` e em diretórios especificados pela variável de ambiente `CDAT_TYPE` (separado por dois pontos). Todos os diretórios que contêm um arquivo `manifest.xml` são considerados como um tipo de coleta válido.

Sintaxe

```
cdat list-types -h
cdat list-types [-v]
```

Sinalizadores

Sinalizador	Descrição
<code>-h</code>	Exibe o uso do comando.
<code>-v</code>	Possibilita saída detalhada e exibe o uso de tipos de coleta.

Saída

Se o subcomando é usado com `-h` ou com um parâmetro inválido, o subcomando imprime a ajuda. Caso contrário, o comando lista os tipos de coleta instalados.

Exemplo

```
$ cdat list-types -v
List of available collect types:

trace (/usr/lib/cdat/types/trace):  \\Records selected system events from nodes of type AIX LPAR or VIOS.
Usage: trace [-d duration] -- [trace_options]
-d duration                          \\duration of collect in seconds (default is 30)
trace_options AIX trace(5) command options

perfpmr (/usr/lib/cdat/types/perfpmr):  \\Retrieves the result of the perfpmr command from nodes of type AIX LPAR.
Usage: perfpmr [-d duration]
-d duration                          \\duration of collect in seconds (default is 600)

snap (/usr/lib/cdat/types/snap):      \\Gathers system configuration information from nodes of type AIX LPAR or VIOS.
Usage: snap [snap_options]
snap_options AIX snap(5) command options (default is -a)
```

Código de Retorno

Se o subcomando for usado com `-h` ou com um parâmetro inválido, o código de retorno será 1. Se o comando não puder listar os tipos de coleta, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Subcomando `cdat archive`

Para criar um archive compactado `tar.Z` de coletas armazenadas no repositório, o subcomando `archive` pode ser usado.

Finalidade

O subcomando `archive` pode ser usado para criar um arquivo compactado `tar.Z` de coletas armazenadas no repositório.

Descrição

O subcomando **archive** pode ser usado para criar um arquivo compactado **tar.Z** de coletas armazenadas no repositório. É possível arquivar todas as coletas associadas a um determinado número do PMR ou a uma coleta especificada pelo nome.

Sintaxe

```
cdat archive -h
cdat
archive [-f File] -p PMR
cdat archive -f File Id
```

Sinalizadores

Sinalizador	Descrição
-h	Exibe o uso do comando.
Id	Especifica o identificador da coleta a ser arquivada.
-p	Especifica o número do PMR de coletas a serem arquivadas.
PMR	
-f	Especifica o nome do archive a ser criado. No caso em que -p é especificado, o nome do arquivo padrão é PMR.tar.Z, PMR é o número do PMR.
ARQUIVO	

Saída

Se for usado com **-h** ou com um parâmetro inválido, o subcomando imprimirá a ajuda. Caso contrário, o comando criará um archive contendo todas as coletas correspondentes ao número do PMR ou ao nome da coleta especificado. Esse archive contém todos os diretórios das coletas e também um arquivo de texto (**README**) que descreve as coletas. Esse arquivo de texto é a saída do comando **cdat show -v** no número do PMR ou no nome da coleta especificado.

Exemplo

```
% cdat archive -p 12345,123,123
-f archive.tar.Z
% uncompress -c archive.tar.Z | tar tf -
README
mycollect/
mycollect/logs.txt
mycollect/trace/
mycollect/trace/fleuret_ios/
mycollect/trace/fleuret_ios/logs.txt
mycollect/trace/fleuret_ios/trcfile
mycollect/trace/fleuret_ios/trcfmt
mycollect/trace/mnffdc1/
mycollect/trace/mnffdc1/logs.txt
mycollect/trace/mnffdc1/trcfile
mycollect/trace/mnffdc1/trcfmt
mycollect/trace/sohmc/
mycollect/trace/sohmc/logs.txt
mycollect/trace/sohmc/errors.txt
mycollect/trace/mnffdc2/
mycollect/trace/mnffdc2/logs.txt
mycollect/trace/mnffdc2/trcfile
mycollect/trace/mnffdc2/trcfmt
```

Código de Retorno

Se usado com **-h** ou com um parâmetro inválido, o código de retorno será 1. Se o comando não conseguir criar um archive, o código de retorno será 2. Caso contrário, o código de retorno será 0.

Coletas Periódicas

É possível planejar coletas de dados periódicas usando o comando **crontab(1)**.

Por exemplo, para executar o tipo de coleta **snap** todo dia à meia noite:

```
% crontab -e cdat
0 0 * * * /usr/bin/cdat collect -q -t snap -f /cdat/nodes.txt
```

Com esta configuração, **cdat** cria um novo diretório em **/cdat** (e um novo ID de coleta) todo dia à meia noite que conterà os dados **snap** para cada nó presente em **/cdat/nodes.txt**. É possível (mas não obrigatório) sobrescrever coletas **snap** anteriores especificando um ID de coleta para o subcomando de coleta usando a opção **-i**:

```
% crontab -e cdat
0 0 * * * /usr/bin/cdat collect -q -t snap -i my_daily_snap -f /cdat/nodes.txt
```

Neste caso, o mesmo diretório é usado para todas as coletas. Apenas os dados **snap** válidos são mantidos para cada nó presente em **/cdat/nodes.txt**. Dados **snap** mais antigos são sobrescritos. Remover coletas planejadas anteriormente pode ser feito executando o comando **crontab -e cdat** e removendo a entrada apropriada do arquivo. Coletas planejadas podem ser gerenciadas de forma transparente a partir de menus SMIT, o que evita a manipulação manual de **crontab**.

Arquivos de Log

Os arquivos de log podem ser usados para diagnosticar problemas encontrados durante uma coleta.

Há dois tipos de arquivos de log, um arquivo de log por coleta que contém informações de sincronização entre nós e um arquivo de log por nó que contém informações sobre fases de coleta e comandos remotos. Um arquivo de log é chamado **logs.txt** e está localizado no diretório de coleta, por exemplo **/cdat/00000001/**, ou o diretório de nós, por exemplo, **/cdat/00000001/trace/node1**, respectivamente, um arquivo de log por coleta e um arquivo de log por nó. Cada linha em um arquivo de log é prefixada com um registro de data e hora do nó central.

Aqui está um exemplo de um *arquivo de log por coleta* para a coleta de rastreo no nó **node1**:

```
% cat /cdat/00000001/logs.txt
2010-07-29 09:17:42: Creating "/cdat/00000001/collect.xml"
2010-07-29 09:17:42: Retrieving node information for node1
2010-07-29 09:17:44: Starting collect type "trace"
2010-07-29 09:17:44: Creating directory "/cdat/00000001/trace"
2010-07-29 09:17:44: Creating directory "/cdat/00000001/trace/node1"
2010-07-29 09:17:44: Starting "check" phase on node1 (LPAR): pid 5570774
2010-07-29 09:17:44: Waiting for children to terminate
2010-07-29 09:17:44: pid 5570774 (node1) terminated with exit status 0
2010-07-29 09:17:44: Starting "init" phase on node1 (LPAR): pid 5570776
2010-07-29 09:17:44: Waiting for children to terminate
2010-07-29 09:17:44: pid 5570776 (node1) terminated with exit status 0
2010-07-29 09:17:44: Starting "execute" phase on node1 (LPAR): pid 5570778
2010-07-29 09:17:44: Waiting for children to terminate
2010-07-29 09:17:46: pid 5570778 (node1) terminated with exit status 0
2010-07-29 09:17:46: Starting "terminate" phase on node1 (LPAR): pid 5570780
2010-07-29 09:17:46: Waiting for children to terminate
2010-07-29 09:17:47: pid 5570780 (node1) terminated with exit status 0
2010-07-29 09:17:47: Starting "grab" phase on node1 (LPAR): pid 5570782
2010-07-29 09:17:47: Waiting for children to terminate
2010-07-29 09:17:49: pid 5570782 (node1) terminated with exit status 0
2010-07-29 09:17:49: Starting "clean" phase on node1 (LPAR): pid 5570784
2010-07-29 09:17:49: Waiting for children to terminate
2010-07-29 09:17:50: pid 5570784 (node1) terminated with exit status 0
```

Aqui está um exemplo de um *arquivo de log por nó* para a coleta de rastreo para o nó **node1**:

```

% cat /cdat/00000001/trace/node1/logs.txt
*** "check" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
*** "init" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
*** "execute" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
Running remote command "LANG=C /usr/sbin/trace -a -o /tmp/cdat.trc " on "node1" as user "cdat"
Return code 0
*** "terminate" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
Running remote command "LANG=C /usr/bin/trcstop" on "node1" as user "cdat"
Return code 0
*** "grab" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
Retrieving /tmp/cdat.trc from LPAR node1 using SCP
Retrieving /etc/trcfmt from LPAR node1 using SCP
*** "clean" phase ***
Running "/usr/lib/cdat/types/trace/trace -d1"
Running remote command "rm -f /tmp/cdat.trc" on "node1" as user "cdat"
Return code 0

```

Tipos de Coleta

A estrutura de Agregação de Dados do Cluster fornece um conjunto de tipos de coleta.

O conjunto de tipos de coleta padrão é: **snap**, **perfpmr** e **trace**.

Tipos de Coleta Padrão

Os tipos de coleta padrão estão a seguir:

- **snap**: Coleta dados **snap** de um AIX LPAR ou VIOS.
- **perfpmr**: Coleta dados **perfpmr** de um AIX LPAR.
- **trace**: Rastreia dados de um AIX LPAR ou VIOS.

Tipo de Coleta Snap

O tipo de coleta **snap** executa o comando **snap** com as opções especificadas em um AIX LPAR ou VIOS e recupera o conteúdo do diretório `/tmp/ibmsupt` no armazenador central. O tipo de coleta **snap** suporta todas as opções suportadas pelo comando AIX **snap**.

Exemplo

Segue um exemplo de uma coleta **snap** (padrão **snap -a**) para dois nós (um para AIX LPAR e outra para VIOS):

```

% cdat collect -t snap -n LPAR:root@lpar1 -n
VIOS:padmin@vios1

```

Tipo de Coleta perfpmr

O tipo de coleta **perfpmr** instala o comando **perfpmr** em um AIX LPAR (por exemplo, `perf61.tar.Z`) e executa-o com o comando `perfpmr.sh 600`.

O tipo de coleta **perfpmr** recupera o arquivo produzido por `perfpmr.sh -o perfddata -z` no armazenador central sob o diretório `perfpmr.pax.gz`. O tipo de coleta **perfpmr** suporta a opção **-d <duration>** que especifica a duração da análise em segundos (padrão de 600 segundos). Se nenhuma versão da ferramenta **perfpmr** corresponder ao nível de Sistema Operacional de um nó remoto (por exemplo, `perf61.tar.Z` para AIX 6.1) sob o diretório `/usr/lib/cdat/types/perfpmr/`, o tipo de coleta **perfpmr** falha para esse nó e você deve fazer download da versão apropriada de **perfpmr** para o nó e copiá-lo no diretório `/usr/lib/cdat/types/perfpmr/`.

Exemplos

Segue um exemplo de tipo de coleta:

```
Phase "check" of collect type "perfpmr" failed for node lpar1:
### BEGIN REASON
/usr/lib/cdat/types/perfpmr/perf61.tar.Z not found.
Please install a version of PERFPMR suitable for AIX 6.1.3.0 under
/usr/lib/cdat/types/perfpmr/perf61.tar.Z.
You may find it at the following URL:
ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr/
### END REASON
```

Segue um exemplo para uma coleta perfpmr (duração igual a 60 segundos) para os dois nós AIX LPAR:

```
% cdat collect -t perfpmr, "-d 60" -n LPAR:lpar1 -n LPAR:root@lpar2
```

Nota: Talvez recupere o pacote perfpmr (por exemplo, perf61.tar.Z) em <ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr> e selecione a versão apropriada (para perf61, faça download de perf61.tar.Z para AIX).

Tipo de Coleta de Rastreio

O tipo de coleta de rastreio executa o comando **trace -a** em um AIX LPAR ou VIOS.

O tipo de coleta de rastreio é suspenso durante o número especificado de segundos e depois executa o comando **trcstop**. Ele recupera o arquivo de rastreio no armazenador central sob o arquivo **trcfile**. O tipo de coleta de rastreio suporta a opção **-d <duration>**, que especifica a duração do rastreio em segundos (padrão de 30 segundos). As opções podem ser passadas ao comando AIX **trace(5)** separadas das opções de script pelo símbolo **--**.

Exemplo

O seguinte comando pode ser executado:

```
% cdat collect -t trace, "-d 60 -- -j 492" -n LPAR:lpar1 \
-n LPAR:root@lpar2
```

Este comando executa o comando **trace** com a opção **-j 492** durante 60 segundos nos nós **lpar1** e **lpar2**

Estendendo a Estrutura

Essa seção descreve como estender a estrutura.

É possível definir um novo tipo de coleta de qualquer uma das seguintes formas:

1. Criando um diretório com o nome do novo tipo de coleta no diretório **/var/adm/ras/cdat/**.
2. Criar um arquivo XML de manifesto no diretório que descreve a função do tipo de coleta recém incluído.
3. Gravando um script no diretório para executar a operação de coleta.

Formato do Arquivo manifest.xml

Um arquivo **manifest.xml** descreve o que é um tipo de coleta e quais opções ele suporta.

Segue um exemplo de um arquivo **manifest.xml** para o tipo de coleta **trace**:

```
<?xml version="1.0"?>
<manifest>
  <description>
    Retrieve trace data from remote nodes.
  </description>
  <script-arg id="d" mandatory="0" default="30"
```

```

    <description>Duration in seconds</description>
  </script-arg>
  <pass-through-arg default="-a">
    <description>AIX trace(5) command options</description>
  </pass-through-arg>
</manifest>

```

Segue a Definição de Esquema XML para o arquivo manifest.xml:

```

<?xml
version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="manifest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="description"/>
        <xs:element name="script-arg">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="description"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required"/>
            <xs:attribute name="mandatory" type="xs:integer" use="required"/>
            <xs:attribute name="default" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="pass-through-arg">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="description"/>
            </xs:sequence>
            <xs:attribute name="default" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Scripts de Coleta Customizados

É possível localizar as informações em um conjunto de seis fases para a operação de coleta: verificar, iniciar, executar, terminar, capturar e limpar que a estrutura de Cluster Data Aggregation Tool define.

Um tipo de coleta pode fornecer um script ou um executável para implementar essas fases. Nem todas as fases são necessárias e se uma determinada fase não for implementada, esta fase será ignorada. Há um ponto de sincronização após cada fase; isto é, a estrutura aguarda que a fase anterior termine em todos os nós antes de iniciar a próxima fase.

- O propósito da fase **verificar** é verificar se a operação de coleta pode ser executada no nó remoto. O processo inclui etapas como verificar o nível de S.O. do nó remoto e verificar o espaço em disco.
- O propósito da fase **iniciar** é configurar o ambiente que é necessário para a execução da coleta. O processo inclui a instalação de conjuntos de arquivo ou scripts no nó remoto. É melhor usar o serviço `push_file` para copiar arquivos no nó remoto.
- O propósito da fase **executar** é iniciar o processo de coleta nos nós remotos.
- O propósito da fase **terminar** é parar o processo de coleta nos nós remotos de forma que o resultado da coleta esteja disponível.
- O objetivo da fase de **captura** é recuperar os dados coletados a partir do nó remoto e copiá-los para o diretório de destino. Use o serviço `get_file` para recuperar arquivos ou diretórios de um nó remoto, porque este serviço gerencia a autenticação com o nó remoto automaticamente e é capaz de estender o tamanho do repositório quando necessário.

- O objetivo da fase de **limpeza** é executar uma limpeza no nó local ou remoto. O processo inclui a remoção de arquivos temporários ou a remoção de conjuntos de arquivos instalados durante a fase de inicialização do nó remoto. Observe que a fase de limpeza é sempre executada, mesmo se a coleta falhar ou for interrompida.

Um script de coleta customizado deve ser fornecido para implementar fases como verificar, iniciar, executar, terminar, capturar e limpar. Não é obrigatório implementar todas as fases. Por exemplo, não é necessário que forneça uma implementação de iniciar ou terminar, se nenhuma ação especial estiver sendo executada durante essas fases. O script é gravado em qualquer linguagem de programação (scripts ou binários compilados). Você chama os serviços `push_file`, `get_file`, `mlog` e `remote_cmd` fornecidos pela estrutura de Cluster Data Aggregation Tool a partir dos scripts customizados para arquivos de transferência, mensagens de log ou executar comandos em um nó remoto.

Cada fase implementada do script customizado deve seguir as regras descritas para os códigos de retorno:

Valor de retorno	Descrição
0	Indica se a fase de coleta foi bem-sucedida.
1	Indica se o nó remoto atual deve ser ignorado em fases subsequentes.
Mais de 128	Indica se ocorreu um erro durante a execução da fase. A estrutura chama a fase limpar quando tal erro ocorre.

Um script de coleta é chamado com as seguintes variáveis de ambiente (configurado pela estrutura):

Nome da variável	Descrição
CDAT_DEST_DIR	Especifica o diretório no nó principal em que os dados coletados no nó remoto estão armazenados.
CDAT_HOST	Especifica o nome do host do nó remoto.
CDAT_PHASE	Especifica a fase a ser executada.
CDAT_PMR	Se configurado, especifica o número de PMR associado à coleta atual.
CDAT_SRVC_DIR	Especifica o caminho no nó principal para os serviços <code>get_file</code> , <code>push_file</code> , <code>remote_cmd</code> e <code>mlog</code> .
CDAT_TYPE	Especifica o tipo de nó (isto é, LPAR, HMC, VIOS ou PSCALE).
CDAT_TYPE_DIR	Especifica o diretório que contém o script para o tipo de coleta atual.
CDAT_USER	Especifica o nome de usuário para efetuar login no nó remoto.

Exemplo

Segue um exemplo de uma definição de um novo tipo de coleta que recupera o conteúdo do arquivo `/var/adm/ras/errlog` de nós remotos (do tipo AIX LPAR):

1. Crie um novo diretório sob o diretório `/var/adm/ras/cdat/`:

```
% mkdir -p /var/adm/ras/cdat/myerrlog
```

2. Criar o arquivo `manifest.xml`:

```
% vi
/var/adm/ras/cdat/myerrlog/manifest.xml
<?xml version="1.0"?>
<manifest>
<description>Retrieve the content of the /var/adm/ras/errlog file.</description>
</manifest>
```

Nota: Também é possível criar arquivos localizados `manifest.xml` incluindo um sufixo local (por exemplo, o arquivo `manifest.fr_FR.xml` para francês).

3. Criar o script que busca o arquivo `errlog`. No exemplo a seguir, apenas a fase de captura é implementada porque não há comando para gerar o arquivo (sem fase executar).

```
% vi /var/adm/ras/cdat/myerrlog/myerrlog
#!/bin/sh
if [ $CDAT_PHASE = "grab" ]; then
```

```

$CDAT_SRVC_DIR/get_file /var/adm/ras/errlog
if [ $? -ne 0 ] ; then
    $CDAT_SRVC_DIR/mlog 0 "Could not retrieve errlog from $CDAT_HOST"
    exit 128
fi
fi
exit 0

```

4. Verifique se o novo tipo de coleta está incluído e detectado, executando o seguinte comando:

```
% cdat list-types
```

O comando antecedente lista todos os tipos de coleta disponíveis.

O comando a seguir recupera o conteúdo do arquivo `/var/adm/ras/errlog`:

```
... myerrlog
(/var/adm/ras/cdat/myerrlog)
```

5. Execute o novo tipo de coleta:

```
% cdat collect -t myerrlog
-n LPAR:root@mylpar1 -n LPAR:root@mylpar2
```

Conceitos relacionados:

“Subcomando `cdat collect`” na página 11

Auxiliares de Estrutura

A estrutura fornece um conjunto de serviços para coletar scripts usados para efetuar log de mensagens, executar comandos em nós remotos ou transferir arquivos entre nós remotos e o nó principal central.

Para ser móvel, scripts de coleta usam esses serviços para realizar aquelas ações em vez de criar uma sua própria. Esses serviços permitem que os scripts de coleta ignorem o protocolo de transporte subjacente, que é usado para conectar a nós remotos como Secure Shell (SSH), Remote Execution Protocol (REXEC), Telnet e Protocolo de Transferência de Arquivos (FTP).

Serviço `remote_cmd`

Use o serviço `remote_cmd` para executar comandos nos nós remotos.

As informações sobre como se conectar com o nó remoto são recuperadas a partir das variáveis de ambiente `CDAT_USER`, `CDAT_HOST` e `CDAT_TYPE`. O serviço `remote_cmd` pode usar SSH, REXEC ou Telnet para executar o comando no nó remoto. Se a criação de log para o nó remoto solicitar uma senha (por exemplo, nenhuma fase de pré-autenticação foi executada para este nó), o serviço `remote_cmd` falhará.

Sintaxe

```
remote_cmd <command>
```

Serviço `push_file`

Use o serviço `push_file` para copiar arquivos ou diretórios do nó principal central em um nó remoto.

As informações sobre como se conectar com o nó remoto são recuperadas a partir das variáveis de ambiente `CDAT_USER`, `CDAT_HOST` e `CDAT_TYPE`. O serviço `push_file` pode usar Secure Copy Protocol (SCP) ou Protocolo de Transferência de Arquivos (FTP) para copiar os arquivos no nó remoto. Se a cópia de um arquivo no nó remoto solicitar uma senha (por exemplo, nenhuma fase de pré-autenticação foi executada para este nó), o serviço `push_file` falhará.

Sintaxe

```
push_file <local
file>....<remote directory>
```

Serviço `get_file`

Use o serviço `get_file` para copiar arquivos ou diretórios do nó remoto para o nó principal central. As informações sobre como se conectar com o nó remoto são recuperadas a partir das variáveis de ambiente `CDAT_USER`, `CDAT_HOST` e `CDAT_TYPE`.

A variável de ambiente `CDAT_DEST_DIR` especifica o local em que os arquivos devem ser copiados. O serviço `get_file` pode usar Secure Copy Protocol (SCP) ou FTP para copiar os arquivos do nó remoto. Se a cópia de um arquivo do nó remoto solicitar uma senha (por exemplo, nenhuma fase de pré-autenticação foi executada para este nó), o serviço `get_file` falhará.

Sintaxe

```
get_file <remote
file>...<local file>
```

Um diretório pode ser especificado em vez de **arquivo local** para recuperar diversos arquivos ou diretórios com uma única chamada.

```
get_file <remote file>...<local
directory>
```

Antes de recuperação o arquivo, o serviço `get_file` determina se há espaço livre no sistema de arquivo de destino para armazenar esse arquivo. Se este não for o caso, o serviço `get_file` estende o tamanho do sistema de arquivos automaticamente se o comando `cdat collect` foi chamado com a opção `-g`.

Serviço `mlog`

Use o serviço `mlog` para fazer log de mensagens de diagnóstico. Essas mensagens de diagnóstico são armazenadas nos arquivos de log por nó.

Sintaxe

```
mlog <level> <message
to log>
```

Um nível de gravidade pode ser especificado com `<level>` (um número inteiro). Se o nível for `-1`, a mensagem também será impressa na saída `stderr` (e no arquivo `collect log`), quando a fase de coleta atual terminar.

Iniciação Rápida CDAT

Antes de usar a Cluster Data Aggregation Tool, você deve inicializá-la no nó principal, o nó central que reúne dados de determinação de problema.

Para inicializar a Cluster Data Aggregation Tool, conclua as etapas a seguir:

1. Efetue login no nó central de Cluster Data Aggregation Tool como raiz e execute o comando `smit cdat`.
2. Selecione **Create the Repository**.
3. Selecione **Discover Nodes** para criar um arquivo que lista todos os nós remotos em que é preciso coletar dados de determinação de problema. O nome padrão para este arquivo é `/cdat/nodes.txt`.
4. Selecione **Manage Remote Nodes**.
5. Selecione **Initialize Access to Remote Nodes**.
6. Especifique o nome do arquivo que lista todos os nós remotos no campo **Node filename**.
7. Pressione **Enter** para executar a inicialização.
8. Responda a todas as perguntas quando solicitado e especifique uma senha para o usuário `cdat` que é criado em cada nó remoto. Você deve fornecer a senha `root` de cada nó remoto.

Após o nó principal de Cluster Data Aggregation Tool ter sido inicializado, você é capaz de coletar dados de determinação de problema nos nós remotos usando tipos de coleta padrão como **perfpmr**, **snap** ou **trace**

Para coletar dados em nós remotos, conclua as etapas a seguir:

1. Efetue login no nó principal de Cluster Data Aggregation Tool como raiz ou usuário **cdat** e execute o comando **smit cdat**.
2. Selecione **Collect Data from Remote Nodes**.
3. Se houver um arquivo que lista todos os nós remotos em que é preciso coletar dados de determinação de problema, use-o para especificar os nós remotos. O valor padrão para o campo **Node filename** é **/cdat/nodes.txt**. Se não tiver um arquivo que lista os nós remotos, insira manualmente os nós remotos no campo **Remote nodes**.
4. Selecione o tipo de coleta que deseja executar no campo **Collect type**.
5. Especifique opções no campo **Parameters**.
6. Pressione **Enter** para executar a operação de coleta.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos neste documento em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um programa, produto ou serviço IBM não significa que apenas o programa, produto ou serviço da IBM possa ser usado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser usado em substituição a este produto, programa ou serviço da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do usuário.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente licença alguma sobre tais patentes. Pedidos de licença 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 IBM em seu país ou envie pedidos de licença, por escrito, para:

Licença de propriedade intelectual
Lei de propriedade intelectual e legal
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica ao Reino Unido nem a nenhum país em que tais disposições não estejam de acordo com a legislação local: 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 estados não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Estas informações podem incluir imprecisões técnicas ou erros tipográficos. São feitas mudanças periódicas nas informações aqui contidas; tais mudanças serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, 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 o uso desses websites é de inteira responsabilidade do Cliente.

A IBM pode usar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão 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 aos termos e condições apropriados, incluindo, em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, a partir 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. As dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente aos seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a mudanças ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Todos os preços IBM mostrados são preços de varejo sugeridos pela IBM, são atuais e estão sujeitos a mudança sem aviso prévio. Os preços do revendedor podem variar.

Estas informações foram projetadas apenas com o propósito de planejamento. As informações aqui contidas estão sujeitas a mudanças antes que os produtos descritos estejam disponíveis.

Estas informações contêm exemplos de dados e relatórios usados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços usados 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, uso, marketing ou distribuição de programas de 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 nenhum tipo. A IBM não deverá ser responsabilizada por quaisquer danos decorrentes do uso dos programas de amostra.

Cada cópia ou parte destes programas de amostra ou qualquer trabalho derivado deve incluir um aviso de copyright com os dizeres:

Partes deste código são derivadas dos Programas de Amostras da IBM Corp.

© Copyright IBM Corp. _insira o ano ou anos_. Todos os direitos reservados.

Considerações sobre política de privacidade

Os Produtos de software IBM®, incluindo soluções de software como serviço, (“Ofertas de Software”) podem usar cookies ou outras tecnologias para coletar informações de uso do produto, para ajudar a melhorar a experiência do usuário final, para customizar as interações com o usuário final ou para outros fins. Em muitos casos, nenhuma informação pessoalmente identificável é coletada pelas Ofertas de Software. Algumas de nossas Ofertas de Software podem ajudar a coletar informações pessoalmente identificáveis. Se esta Oferta de Software usar cookies para coletar informações pessoalmente identificáveis, informações específicas sobre o uso de cookies desta oferta serão definidas 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 pessoalmente identificáveis de usuários finais via cookies e outras tecnologias, você deve buscar seu próprio aconselhamento jurídico sobre quaisquer leis aplicáveis a tal coleta de dados, incluindo requisitos para aviso e consento.

Para obter mais informações sobre o uso de várias tecnologias, incluindo cookies, para estes fins, consulte a Política de Privacidade da IBM em <http://www.ibm.com/privacy> e Declaração de Privacidade Online da IBM na <http://www.ibm.com/privacy/details> 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>.

Marcas comerciais

IBM, o logotipo IBM e [ibm.com](http://www.ibm.com) são marcas comerciais ou marcas registradas da International Business Machines Corp., registradas em vários países no mundo todo. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas. Uma lista atual de marcas comerciais da IBM está disponível na web em Copyright and trademark information em www.ibm.com/legal/copytrade.shtml.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Índice Remissivo

Caracteres Especiais

/cdat/nodes.txt 15, 21
/usr/lib/cdat/types/ 13

A

access 1, 2, 9, 10
archive 1, 2, 13
arquivo de log 15
Auxiliares 20

C

captura 1, 17, 18
cdat 2, 3, 10, 15, 21
CDAT_DEST_DIR 21
CDAT_HOST 20, 21
CDAT_TYPE 20, 21
CDAT_USER 20, 21
cdat.xml 7, 8
check 1, 7, 17, 18
coletar 1, 2, 11, 21
collect.xml 11

D

discover-node 1, 2
discover-nodes 9
dump 1

E

estendendo estrutura 17, 18
excluir 1, 2, 8
execução 1, 17, 18
executar 9

G

get_file 17, 18, 21

H

HMC 1, 9, 10
hscroot 9

I

Iniciação Rápida 21
inicialização 1, 2, 3, 9, 17, 18
iptrace 1
IVM 9

L

limpo 1, 17, 18
list-nodes 1, 2, 10

list-types 1, 2, 13
log de coleta 21
LPAR 1, 9, 16, 17
lpar1 17
lpar2 17

M

manifest.xml 13, 17, 18
mlog 17, 18, 21

N

Nó 10
nodes.txt 9, 10
nós remotos 11
número do PMR 8

P

pacote perfpmr 16
padmin 9, 10
perfpmr 1, 16, 21
push_file 17, 18, 20

R

raiz 3, 21
remote_cmd 17, 18, 20

S

show 1, 2, 4, 13
SMIT 1
smit cdat 21
snap 1, 15, 16, 21
stderr 21

T

término 1, 17, 18
Tipos de Coleta Padrão 16
trace 1, 16, 17, 18, 21
trcfile 17

U

Usuário 10

V

var/adm/ras/cdat/ 13
VIOS 1, 9, 16, 17



Impresso no Brasil