

Zowe CLI e VSAM

Selecionando registros com Zowe CLI e VSAM



15 passos



120 minutos

ZXPL> Zowe

DESCRIPTION

Welcome to Zowe CLI!

1. Instalando o Zowe CLI

Estamos usando o plug-in Zowe Explorer para VS Code ao longo deste concurso, mas o Zowe faz muito mais e é responsável por trazer muito mais para o mainframe.

Para deixar claro, **você está instalando o Zowe CLI em seu próprio computador, não no mainframe**. Você usará o Zowe CLI para fazer uma interface com o Zowe e o z/OSMF que está sendo executado no mainframe, mas você estará conduzindo a maior parte desse desafio em seu próprio computador.

Os usuários do Linux podem precisar explorar um pouco para encontrar o que funciona em seu sistema específico, mas deve se aproximar das etapas do Mac, apenas substituindo o arquivo de perfil do shell correto.



O DESAFIO

Você interage com o mainframe por meio de uma série de transações. Você emite um pedido para visualizar os trabalhos, outro para visualizar conjuntos de dados, outro para emitir um comando. Nos bastidores, o *framework open source*, Zowe, está trabalhando para vincular os recursos do mainframe a APIs, comandos e bibliotecas fáceis de usar. Simplificando, você pode acessar um sistema Z de praticamente qualquer lugar, usando uma ampla variedade de ferramentas e plataformas.

ANTES DE VOCÊ COMEÇAR

Este desafio fará mais sentido se você já tiver concluído todos os desafios do Fundamentals, pois ele usa um pouco de tudo a partir daí. Não há requisito, mas faremos suposições sobre o que você sabe neste momento.

```
> ssh2@1.4.0 install /Users/joris/.npm-global/lib/node_modules/@zowe/cli/node_modules/ssh2
> node install.js

  CXX(target) Release/obj.target/sshcrypto/src/binding.o
  SOLINK_MODULE(target) Release/sshcrypto.node
Succeeded in building optional crypto binding

> @zowe/cli@6.36.1 postinstall /Users/joris/.npm-global/lib/node_modules/@zowe/cli
> node ./scripts/validatePlugins && node ./scripts/printSuccessMessage

Since you re-installed Zowe CLI, we are re-validating any plugins.

----- Validation results for plugin '@zowe/secure-credential-store-for-zowe-cli' -----
This plugin was successfully validated. Enjoy the plugin.

  Zowe CLI has been successfully installed. You can safely ignore all non-plug-in
  related errors and warnings. Please check above for any plug-in related issues.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: cpu-features@0.0.2 (node_modules/@zowe/cli)
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: cpu-features@0.0.2 install: `node-gyp rebu...
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Exit status 1

+ @zowe/cli@6.36.1
added 224 packages from 162 contributors in 47.629s
```

```
PS C:\Users\JeffreyBisti> cmd
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.
```

```
C:\Users\JeffreyBisti>npm i -g @zowe/cli
C:\Users\JeffreyBisti\AppData\Roaming\npm\bright -> C:\Users\JeffreyBisti\AppData\Roaming\npm
  modules\@zowe\cli\lib\main.js
C:\Users\JeffreyBisti\AppData\Roaming\npm\zowe -> C:\Users\JeffreyBisti\AppData\Roaming\npm\m
  odules\@zowe\cli\lib\main.js

> @zowe/cli@6.22.0 postinstall C:\Users\JeffreyBisti\AppData\Roaming\npm\node_modules\@zowe\cli
> node ./scripts/validatePlugins
```

```
Since you re-installed Zowe CLI, we are re-validating any plugins.
No plugins have been installed into your CLI application.
+ @zowe/cli@6.22.0
updated 4 packages in 14.432s
```

```
C:\Users\JeffreyBisti>zowe
```

3. Configuração do NPM para Windows

No Windows, primeiro vamos mudar do PowerShell para o CMD e, em seguida, instalar o zowe cli usando npm, o gerenciador de pacotes do Node. Isso deve funcionar para a maioria dos usuários, embora sua saída possa parecer um pouco diferente do que você vê na imagem de captura de tela.

- 1: type cmd (this will change the shell to cmd from PowerShell)
- 1: npm i -g @zowe/cli
- 2: zowe

Continua travado? Entre nos fóruns para obter orientação.

DESCRIPTION

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simplified way to interact with IBM z/OS.

For additional Zowe CLI documentation, visit <https://zowe.github.io/zowe-cli-docs/>

For Zowe CLI support, visit <https://zowe.org>

USAGE

`zowe <group>`

Where `<group>` is one of the following:

4. Um comando de quatro letras

Agora que estamos com tudo configurado, abra um novo terminal e digite o comando **zowe**. Assim mesmo, desse jeito.

Certifique-se de seguir as instruções de configuração de perfil das etapas 5 a 6 do desafio REXX1, caso contrário, isso poderá falhar.

Você receberá uma descrição, uma lista de grupos de comandos e opções. Vamos gastar muito desse desafio passando por esses grupos de comando, e muitos deles devem soar um tanto quanto familiares. Acesse zowe.org para saber mais.

"ME FALA MAIS SOBRE ZOWE. ISSO É COISA DA IBM OU...?"

Zowe é um projeto de código aberto para z/OS, destinado a tornar a plataforma mais acessível para usuários que não estão começando com anos e anos de experiência em mainframe. O projeto Zowe contém contribuições de indivíduos e empresas da comunidade de mainframe. Isso inclui o plug-in VS Code, várias APIs e o Zowe CLI que você está prestes a explorar.

Zowe é um projeto do [Open Mainframe Project](#), que é um projeto gerenciado pela [Linux Foundation](#). Não é um produto IBM, embora a IBM seja uma colaboradora, suporte e continue a defender o Zowe como um modelo estratégico para trazer novos recursos e usuários para a plataforma de mainframe.

Uma das melhores maneiras de se conectar com empregadores e pessoas informadas é prestar atenção ao que está acontecendo nessas comunidades e ajudar sempre que encontrar uma oportunidade.

USAGE

`zowe zos-console <group>`

Where `<group>` is one of the following:

GROUPS

`collect` Collect z/OS console command responses
`issue` Issue z/OS Console Commands

GLOBAL OPTIONS

`--response-format=json` | `--rfj` (boolean)

Produce JSON formatted data from a command

`--help` | `-h` (boolean)

Display help text

`--help-examples` (boolean)

EXAMPLES

– Submit the JCL in the data set "ibmuser.cntl(deploy)":

`$ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)"`

– Submit the JCL in the data set "ibmuser.cntl(deploy)", wait for the job to complete and print all output from the job:

`$ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)" --vasc`

– Submit the JCL in the file "iefbr14.txt":

`$ zowe zos-jobs submit local-file "iefbr14.txt"`

– Download all the output of the job with job ID JOB00234 to an automatically generated directory.:

`$ zowe zos-jobs download output JOB00234`

– View status and other details of the job with the job ID JOB00123:

6. Exemplos são boas documentações

Use o comando **zowe zos-jobs --help-examples** para obter uma boa listagem de comandos zowe que você pode usar relacionados a tarefas do z/OS. A saída vai além do que é capturado na captura de tela acima, e há muitas variações disponíveis.

Estamos começando com o básico, não se preocupe, isso ficará um pouco mais empolgante em apenas mais algumas etapas.

```
Monorail:~ jbisti$ zowe zos-tso issue command "status" --rfj
{
  "success": true,
  "exitCode": 0,
  "message": "",
  "stdout": "IKJ56455I Z99999 LOGON IN PROGRESS AT 08:41:30 ON JULY 14, 2020\nIKJ56951I NO BROADCAST MESSAGES\nIKJ56216I NO JOBS FOUND\nREADY \n",
  "stderr": "",
  "data": {
    "success": true,
    "startResponse": {
      "success": true,
      "zosmfTsoResponse": {
        "servletKey": "Z99999-78-aabeaaaq",
        "queueID": "1507336",
        "sessionID": "0x4E",
        "ver": "0100",
        "tsoData": [
          {
            "TSO MESSAGE": {
              "VERSION": "0100",
              "DATA": "IKJ56455I Z99999 LOGON IN PROGRESS AT 08:41:30 ON JULY 14, 2020"
            }
          },
          {
            "TSO MESSAGE": {
              "VERSION": "0100",
              "DATA": "IKJ56951I NO BROADCAST MESSAGES"
            }
          }
        ]
      }
    }
  }
}
```

7. Formatar para JSON

Certifique-se de ter seus jobs ativos e digite o comando **zowe zos-jobs list jobs**

Você recebe de volta uma lista de jobs ativos em execução do z/OS aos quais você tem acesso para consultar. Certinho!

Agora, emita o mesmo comando com **--rfj** (*Response Format JSON*) depois dele. Agora você obtém um output COMPLETO, e a saída está no formato JSON, que pode ser muito mais facilmente interpretado por programas que lidam com o formato JSON.

E o JSON

Por que precisamos de JSON quando a saída original fez todo o sentido para nós?

JSON significa JavaScript Object Notation, e é apenas uma maneira de aninhar os atributos de algo em um objeto para que possa ser totalmente representado sempre que for acessado. Ele tende a ser um pouco mais leve e flexível do que outro formato de arquivo com um objetivo semelhante que você já deve ter ouvido falar, chamado XML.

Em muitas linguagens de programação, você pode simplesmente carregar um objeto JSON e, em seguida, usar a *dot notation* (notação de ponto) para acessar os vários atributos desse objeto JSON, economizando tempo valioso ao programar, em comparação com a tarefa manual de escrever analisadores para extrair informações da saída regular.

```
MTM> zowe zos-files list ds Z99999.ZOWEPS -a
dsname: Z99999.ZOWEPS
blkSz: 6160
catnm: MASTERV.CATALOG
cdate: 2020/08/28
dev: 3390
dsorg: PS
edate: ***None***
extx: 1
lrecl: 80
migr: NO
mvol: N
ovf: NO
rdate: ***None***
recfm: FB
sizex: 15
spacu: CYLINDERS
used: 0
vol: VPWRKA
vols: VPWRKA
```

8. Alocar e listar

Vamos colocar isso em uso. Dê uma olhada no grupo de comandos *files* e use-o para alocar (criar) um data set sequencial chamado **Zxxxxx.ZOWEPS** (com seu próprio ID de usuário, é claro).

Em seguida, use outro comando zowe cli para mostrar os atributos do data set que você acabou de criar. Eles devem ser semelhantes ao acima.

Se você receber uma mensagem de tempo limite, tente adicionar **--responseTimeout 30** ao final do comando, para permitir atrasos na resposta.

```
- dsname: Z99999.ZOWEPS
blkSz: 9600
catnm: CATALOG.ZOS1
cdate: 2020/07/14
dev: 3390
dsorg: PS
edate: ***None***
extx: 1
lrecl: 120
migr: NO
mvol: N
ovf: NO
rdate: ***None***
recfm: FB
sizex: 15
spacu: CYLINDERS
used: 0
vol: VPWRKB
vols: VPWRKB
```

9. Completamente personalizado

Então agora você conhece outra maneira de criar e analisar data sets. O problema é que fizemos isso usando um data set com valores padrão, e uma das grandes vantagens dos data sets é como eles são personalizáveis. Exclua esse conjunto de dados (com outro comando *files*) e use a ajuda (ou a [documentação oficial em zowe.org](#)) para **recriar esse data set sequencial** com alguns atributos personalizados.

Primeiro, queremos que o comprimento do registro (*record length*) seja 120 em vez do padrão 80 (temos alguns registros longos) e queremos um tamanho de bloco (*block size*) de 9600.

Ao obtê-lo, você verá uma leitura diferente para o tamanho do bloco e o comprimento do registro (LRECL), como na captura de tela acima.

EXAMPLES

- Create a VSAM data set named "SOME.DATA.SET.NAME" using default values of INDEXED, 840 KB primary storage and 84 KB secondary space.


```
$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME
```
- Create a 5 MB LINEAR VSAM data set named "SOME.DATA.SET.NAME" with 1 MB of secondary space. Show the properties of the data set when it is created:


```
$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --data-space-size 5MB --secondary-space 1MB --show-attributes
```
- Create a VSAM data set named "SOME.DATA.SET.NAME", which is retained for 100 days:


```
$ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --retention-period 100
```

10. A chave para nossos dados

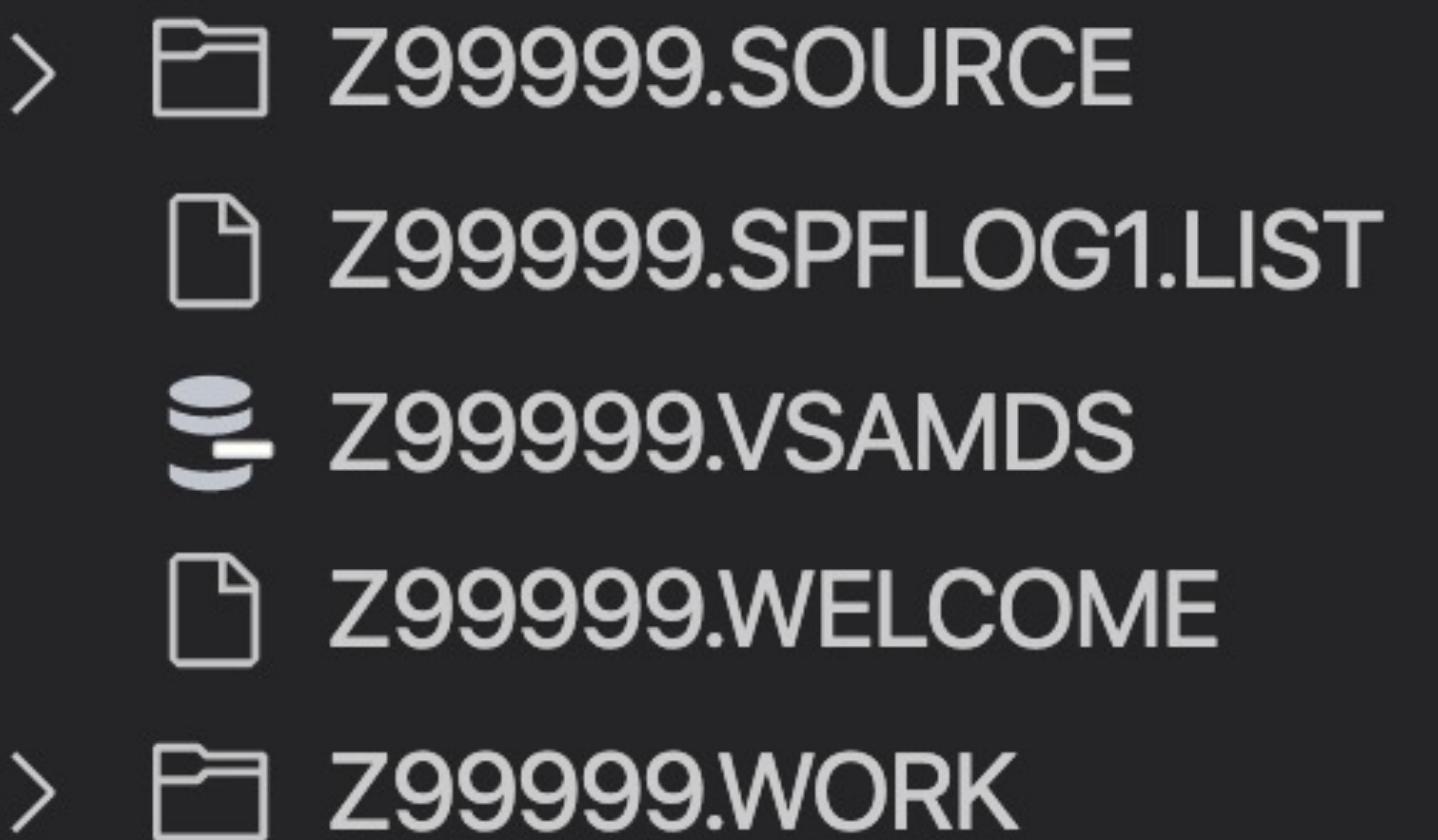
Um tipo de data set que você viu nos menus do Zowe é o VSAM e merece atenção especial. O **VSAM** não é usado para armazenar mensagens JCL ou “Bem-vindo ao Mainframe”. Sua hora de brilhar é quando um aplicativo precisa acessar os registros da forma mais rápida e eficiente possível. Na verdade, sem um software especial para interpretar arquivos VSAM, você não pode abri-los em um editor normal, mas os aplicativos se alimentam desses arquivos com prazer.

É tudo uma questão de eficiência no acesso a dados. Leia mais abaixo.

Aqui estou. Aloque-me como um VSAM

VSAM é complicado e esta pequena caixa cinza não lhe dará anos de experiência trabalhando com data sets VSAM, mas lhe dirá que se você quiser esse trabalho de mainframe, faça toda a leitura e prática com conjuntos de dados VSAM que você puder. Eles são um componente central de qualquer grande empresa que usa mainframe.

Por enquanto, saiba que existem quatro tipos principais de conjuntos de dados VSAM, KSDS (sequência de chave), ESDS (sequência de entrada), RRDS (registro relativo) e Linear (LDS). KSDS e ESDS são os mais comuns, e a diferença se resume a como cada registro é armazenado e acessado. KSDS significa que você faz referência a uma chave (como procurar um número de conta) e obtém as informações dessa conta como registro. O ESDS armazena dados em uma ordem sequencial, para dados que provavelmente serão lidos um após o outro em uma ordem específica. Isso é o suficiente por enquanto, mas se você ainda estiver com sede de saber, [aqui há mais conteúdo a considerar](#).



11. Construa um data set VSAM

Você está mandando bem em alocar data sets. Faça um data set VSAM chamado Zxxxxx.VSAMDS . Consulte a [ajuda online do Zowe](#) para obter uma orientação para o comando.

Quando terminar, observe seus atributos (você sabe como) e notará algo bem interessante; parece que existem TRÊS data sets aqui. Além disso, se você carregá-lo no VS Code, verá um novo ícone elegante. Curioso ainda? Vamos prosseguir.

```

1 IDCAMS SYSTEM SERVICES
0
REPRO -
INFILE(INPUT) -
OUTDATASET(Z99999.VSAMDS)-
ERRORLIMIT(6)
0IDC0005I NUMBER OF RECORDS PROCESSED WAS 1000
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
  
```

12. Carregue-o com registros

Agora, vamos carregá-lo com registros. Você pode usar os dados em **ZXP.PUBLIC.SAMPDATA** ou pode se divertir e fazer os seus próprios. O [Mockaroo.com](#) tem um bom gerador de dados que você pode experimentar, embora com algumas ressalvas:

- 1) A primeira coluna deve estar em ordem
- 2) Omita quaisquer registros/linhas em branco
- 3) Você vai querer zeros à esquerda para as chaves, caso contrário, pode não vê-los em ordem quando você for importar.

Em seguida, localize o arquivo **REPRO** JCL no **ZXP.PUBLIC.JCL** e baixe-o para sua estação de trabalho pessoal, colocando-o na pasta ou diretório em que você está trabalhando no momento. Você terá que ajustar o conjunto de dados de entrada para apontar para seu conjunto de dados de origem. Renomeie o arquivo **REPRO** para **repro.txt**

Use **zowe jobs submit lf "repro.txt"** para enviar o JCL diretamente de sua máquina, através do Zowe CLI. Você receberá uma pequena animação e um *job number*. Verifique esse *job number* e certifique-se de que funcionou sem problemas.

zos

z/OS DFSMS Access Method Services Commands

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

REPRO

z/OS DFSMS Access Method Services Commands

SC23-6846-01

The REPRO command performs the following functions:

- Copies VSAM and non-VSAM data sets. ➤ If the data set is a version 2 PDSE with generations, only the current generation of each member is copied. «
- Copies catalogs
- Copies or merges tape volume catalogs
- Splits integrated catalog facility catalog entries between two catalogs
- Splits entries from an integrated catalog facility master catalog into another integrated catalog facility catalog
- Merges integrated catalog facility catalog entries into another integrated catalog facility user catalog.

13. Vamos preparar o inventário

Vamos falar sobre o que acabamos de fazer. O JCL roda o IDCAMS, que é usado principalmente para gerenciar conjuntos de dados VSAM. No IDCAMS, estamos usando o comando REPRO para carregar um data set sequencial em um data set formatado em VSAM. Há MUITA complexidade acontecendo aqui que não vemos, mas, como antes, há muitas oportunidades de indicar exatamente como você deseja que essa cópia aconteça, incluindo parâmetros criptográficos.

Os dados são os mesmos, mas agora são estruturados fundamentalmente diferentes, indexados por chave e podem ser referenciados com muito mais eficiência por programas (incluindo aqueles escritos em REXX)

Na realidade, esses dados não são muito bem indexados, pois cada linha é sua própria chave, mas se nos aprofundarmos nos detalhes da construção de um cluster VSAM, você poderá ver como as chaves e o tamanho do registro podem ser especificados.

LISTING OF DATA SET -Z99999.VSAMDS

KEY OF RECORD	NAME	NAME	NAME	NAME	NAME	NAME
001354719770	HUBERT	DEMONGEOT	87-8997183	#D230E7		
001354719770	HUBERT	DEMONGEOT	87-8997183	#D230E7	1GKMCC34AR94	
KEY OF RECORD	001359581404	AGNESE	FARRANCE	56-4110060	#9A9D61	
001359581404	AGNESE	FARRANCE	56-4110060	#9A9D61	SCFAB01A76G165	
KEY OF RECORD	001362199763	STEPHEN	TODHUNTER	79-5179893	#906724	
001362199763	STEPHEN	TODHUNTER	79-5179893	#906724	WBAUT9C57BA3	
KEY OF RECORD	001369213008	REAGEN	MCILWRICK	01-1405738	#619A1B	
001369213008	REAGEN	MCILWRICK	01-1405738	#619A1B	JTEBU5JRXF518	
KEY OF RECORD	001384380151	BENNY	LAMBIS	83-6731093	#586AEC	2
001384380151	BENNY	LAMBIS	83-6731093	#586AEC	2C3CCAEG5FH726459	
KEY OF RECORD	001398310239	RAHAL	PENNYCORD	14-4881973	#03E5B3	
001398310239	RAHAL	PENNYCORD	14-4881973	#03E5B3	WBAVC73508A963	
KEY OF RECORD	001399406486	ARIDATHA	TOSELAND	01-9975566	#69E914	
001399406486	ARIDATHA	TOSELAND	01-9975566	#69E914	JM1GJ1T68E11	
KEY OF RECORD	001401405570	ORALLE	KIMMINS	66-8864767	#6198F7	
001401405570	ORALLE	KIMMINS	66-8864767	#6198F7	SAJWA4GB7EL9541	
KEY OF RECORD	001409084356	LYNNE	COLLCOTT	94-7549269	#5B0003	
001409084356	LYNNE	COLLCOTT	94-7549269	#5B0003	1B3CC5FB8AN6801	
KEY OF RECORD	001413762379	POUVTN	APNALDT	95-6304813	#02AE515	

14. Imprimindo registros

amos usar mais um comando IDCAMS para examinar nosso output, o comando chamado PRINT, confira este exemplo (dica) e preste atenção no parâmetro CHARACTER (dica dica) e você quiser que seu output se pareça com a captura de tela cima. Você reunirá informações de várias fontes aqui, então pense no que você tem e no que deseja. Você deseja imprimir esse data set VSAM em formato de caractere. Isso ajuda? Não tenha medo de parar pelos fóruns para alguma ajuda.

OM TRABALHO! VAMOS RECAPITULAR

Você veio para este desafio provavelmente sem saber muito sobre ZCLI, e está saindo sabendo não apenas como se locomover no Zowe CLI, mas um pouco sobre VSAM e IDCAMS.

Você provavelmente também notou o nível de instrução
meçando a mudar de "aqui está um comando" para "descobrir
isso". Bem-vindo às grandes ligas, é assim que funciona agora.

15. Faça valer a pena

Venha para o Zowe CLI, fique para o VSAM e IDCAMS.
Para completar isso, esperamos por 3 coisas:

- 1) Seu data set sequencial **Zxxxxxx.ZOWEPS**
 - 2) Seu data set VSAM **Zxxxxxx.VSAMDS**
 - 3) As primeiras 20 linhas de output do seu PRINT copiadas / coladas em um data set sequencial **Zxxxxxx.OUTPUT.VSAMPRNT**.
Não escreva este data set diretamente do seu JCL, use SYSPRINT e copie/cole as linhas 1-20 do seu SYSPRINT. Precisamos do cabeçalho. Consulte a captura de tela acima como um exemplo (levemente editado).

Quando terminar, envie o **CHKAZCLI**

A SEGUIR...

Agora que o Zowe CLI está em seu kit de ferramentas e você tem alguma experiência básica em VSAM, vamos dar mais alguns detalhes sobre como o VSAM funciona.