

Sumário

Introdução à linguagem de controle de trabalho.....	4
O que é um JCL:.....	4
Vamos entender um JCL:.....	4
O que um JCL faz?.....	5
O que pode ser feito com JCL?.....	5
O que é um JOB?.....	5
JES2 - Fluxo de trabalho:.....	6
Estrutura das declarações JCL.....	6
Sintaxe do Comando.....	6
Cada instrução JCL é dividida em vários campos.....	6
Sintaxe ou formato:.....	7
Geral.....	7
1ª e 2ª colunas.....	7
3ª coluna.....	7
Cartão de Identificação.....	8
4ª até 11ª coluna.....	8
Tipo de Cartões.....	9
12ª até 16ª coluna.....	9
16ª até 71ª coluna.....	9
Parâmetros do JCL.....	12
Cartão: JOB.....	12
ACCOUNT.....	12
REGION.....	12
CLASS.....	12
MSGLEVEL.....	12
TIME.....	13
TYPRUN.....	13
LINES.....	14
Cartão: JOBLIB.....	14
Cartão: EXEC.....	14
PGM.....	14

PROC.....	14
COND.....	14
REGION.....	15
Sub-cartão: STEPLIB.....	15
Sub-cartão: DD.....	15
DSN.....	15
DISP.....	15
UNIT.....	16
SPACE.....	16
DCB.....	16
DSORG.....	17
Subcartão - DD *.....	18
Atributo Dummy.....	18
//SYSIN DD *.....	18
//SYSUDUMP DD SYSOUT=*.....	18
//SYSABEND DD SYSOUT=*.....	18
//SYSMDUMP DD SYSOUT=*.....	18
//CEEDUMP DD SYSOUT=*.....	18
//SYSABOUT DD SYSOUT=*.....	18
JCL - Utilitario.....	20
IEBCOPY.....	20
IEBGENER.....	20
IEHLIST.....	20
IEBCOMPR.....	21
IEBEDIT.....	21
IEHPROGM.....	21
DFSORT & SYNCSORT.....	21
ICETOOL.....	22
IDCAMS.....	22

Introdução à linguagem de controle de trabalho

O que é um JCL:

A linguagem de controle de tarefa, também conhecida como JOB Control Language (JCL) é um conjunto de instruções que você codifica para informar ao sistema operacional z/OS, o JOB que deverá ser executado no JES2 e armazenado no SPOOL, para consulta via SDSF, ele é composto por um cartão de identificação e N cartões de execução.

Vamos entender um JCL:

Para o sistema z/OS funcione para nós, devemos descrever ao sistema, as inúmeras tarefas e atividades que devem estar executando, algo similar ao antigo Config.Sys e Autoexec.Bat do MS-DOS.

Com a IPL executada, todos os serviços ativos, passamos a segunda parte, um JOB para atender a uma demanda do usuário, assim o trabalho que queremos e os recursos que precisamos serão apontados e usamos o Job Control Linguagem (JCL) para fornecer essas informações, criando um script de comandos.

Job Control Language (JCL) é uma linguagem interpretada que direciona o z/OS no manuseio de programas aplicativos, reservando espaço em disco, tempo de cpu e memória RAM.

Em resumo é um conjunto de instruções de controle que fornecem as especificações necessárias para processar um JOB.

- Trabalho: Execução de um ou mais programas relacionados em sequência, também conhecidos como STEPs.
- Etapa do trabalho: Cada programa a ser executado num STEP, bem como suas dependências, seja parâmetros, arquivos, bibliotecas etc.

O que um JCL faz?

JCL costumava se comunicar com o z/OS. sobre seus requisitos para executar um trabalho, associados a inúmeros aplicativos que auxiliaram nesta demanda, exemplo JES2, RACF, DB2, QSAM, VSAM, Gerenciador de Disco etc.

- RACF: Diz ao z/OS quem você é (permite distinções de prioridade), segurança de acesso e caminho de processamento.
- JES2: Diz ao z/OS suas necessidades de espaço em disco DASD e tempo de execução na CPU.
- LOADLIB: Diz ao z/OS quais programas (módulos de carregamento) serão usados no STEP, 1 por Step.
- DB2 : Diz ao z/OS quais conjuntos de dados são necessários para o(s) programa(s) plano de acesso PLAN.
- QSAM e VSAM : Diz ao z/OS quais conjuntos de dados são necessários para o(s) programa(s)

O que pode ser feito com JCL?

1. Envia um JOB para o JES2 executar no z/OS.
2. Solicitar os recursos necessários para executar o trabalho.
3. Controle o processamento dos sistemas do trabalho.
4. Você pode passar valores de parâmetros por meio de JCL para o programa aplicativo usando o parâmetro PARM ou cartão SYSIN na STEP de instrução EXEC .

O que é um JOB?

Um JOB é definido como a execução de um ou mais programas relacionados em uma sequência. Ele fornece a comunicação entre um programa aplicativo (Exemplo: COBOL, Assembler, REXX, Easytreav, Natural Batch ou PL/I) e o z/OS.

JES2 - Fluxo de trabalho:

Durante a execução, um JOB passa pelas seguintes fases:

1. Fila de Entrada
 1. Conversão
 2. Validação
2. Fila de Execução
3. Fila de Saída

Podendo ser consultado a todo momento no SDSF do TSO, desde que seu perfil de usuário tenha acesso permitido pelo RACF.

Estrutura das declarações JCL

Sintaxe do Comando.

As instruções JCL são codificadas em 80 bytes/colunas remetendo aos antigos cartões perfurados.

Porém o JCL utiliza apenas as primeiras 72 das 80 colunas estão disponíveis para codificar JCL.

As últimas 8 colunas são reservadas para um número de sequência opcional e é recomendado não usa-las.

Cada instrução JCL é dividida em vários campos.

```
VIEW      KC02746.WORKBOOK.JCL(JCLCBL63) - 01.03          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
=COLS>  --+---1---+---2---+---3---+---4---+---5---+---6---+---7--
*****  ***** Top of Data *****
000100 //CATCOB63 JOB ('00'),TSO.&SYSUID,NOTIFY=&SYSUID,
000200 //          MSGCLASS=T,MSGLEVEL=1,CLASS=A,REGION=0M
000300 //*-----*
000400 //* COMPILA E LINKEDITA PROGRAMA COBOL BATCH VERSAO 6.3
000500 //*-----*
```

Sintaxe ou formato:

Geral

1ª e 2ª colunas

1. Campo Identificador:

O campo identificador identifica um registro como uma instrução JCL.

Para a maioria das instruções JCL, o campo identificador ocupa as duas primeiras posições de caracteres e deve conter duas barras (/). Existem duas exceções.

Primeiro, a instrução delimitadora possui uma barra na coluna 1 e um asterisco na coluna 2 (/).

3ª coluna

2. Comentário Opcional

Segundo, o campo identificador para uma instrução de comentário tem três caracteres: as duas primeiras colunas contêm barras e a terceira contém um asterisco (/).

```
VIEW      KC02746.WORKBOOK.JCL(JCLCBL63) - 01.04      Columns 00001 00072
Command ==> _____ Scroll ==> CSR_
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
005400 //
005500 /*
005600 /*
005700
```

Cartão de Identificação

4ª até 11ª coluna

3. Campo Nome:

Segue o campo identificador. O campo name associa um nome a uma instrução JCL.

Não é obrigatório em todos os lugares; mas é sempre obrigatório em uma instrução JOB, cartão EXEC e cartão DD, onde fornece um nome para um trabalho.

Um campo de nome deve começar na coluna 3, logo após o campo de identificador //.

É composto por um a oito caracteres, que podem ser letras, números ou caracteres nacionais (#, @ e \$). Além disso, o primeiro caractere de um nome deve ser uma letra ou um caractere nacional.

```
File Edit Edit Settings Menu Utilities Compilers Test Help
VIEW      KC02746.WORKBOOK.JCL(JCLEXEC2) - 01.14      Columns 00001 0007
Command ==>                                     Scroll ==> CSB
=COLS> -----1-----2-----3-----4-----5-----6-----7
***** ***** Top of Data *****
000001 //CATPGM1 JOB ('00'),TSO,4SYSUID,
000002 //          REGION=0M,NOTIFY=4SYSUID,
000003 //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004 //          *-----*
000005 //JOB LIB DD DSN=KC02746.LIB.USERLOAD,DISP=SHR
000006 //STEP01 EXEC PGM=COBOL021
000007 //* ARQUIVO DE ENTRADA *
000008 //INPUTFL DD DSN=KC02746.VAGNER.FB00QSAM,DISP=SHR
000009 //SYSTSPRT DD SYSOUT=*
000010 //SYSPRINT DD SYSOUT=*
000011 //CEEDUMP DD SYSOUT=*
000012 //SYSUDUMP DD SYSOUT=*
000013 //SYSOUT DD SYSOUT=*
000014 //REPORT DD SYSOUT=*
000015
```


Tipo de Cartões

12ª até 16ª coluna

O campo de operação segue o campo de nome e especifica a função das instruções.

Statements do JCL : JOB, EXEC e DD são os diferentes campos de operação.

- JOB : identifica os parâmetros iniciais do cartão JCL
- EXEC : identifica o programa executável que será utilizado esta associado há um Step específico.
- DD : identifica os arquivos de Input, Output utilizados no Step corrente declarado em conjunto com um cartão EXEC.
- 1 byte em branco para separar dos diversos parâmetros.

```
VIEW      KC02746.WORKBOOK.JCL(JCLEXEC2) - 01.14      Columns 00001 00072
Command ==>                                         Scroll ==> CSR
=COLS> -----1-----2-----3-----4-----5-----6-----7-----
***** Top of Data *****
000001 //CATPGM1 JOB ('00'),TSO.&SYSUID,
000002 //          REGION=0M,NOTIFY=&SYSUID,
000003 //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004 //*****
000005 //JOBLIB DD DSN=KC02746.LIB.USERLOAD,DISP=SHR
000006 //STEP01 EXEC PGM=COBOL021
000007 //* ARQUIVO DE ENTRADA *\\
000008 //INPUTFL DD DSN=KC02746.VAGNER.FB80QSAM,DISP=SHR
000009 //SYSTSPRT DD SYSOUT=*
000010 //SYSPRINT DD SYSOUT=*
000011 //CEEDUMP DD SYSOUT=*
000012 //SYSUDUMP DD SYSOUT=*
000013 //SYSOUT DD SYSOUT=*
000014 //REPORT DD SYSOUT=*
000015
000016
000017
```

3. Campo de Operação:

Este campo pode ser codificado em qualquer lugar da JCL, mas deve ser separado do campo nome por pelo menos um espaço em branco.

Porém, convencionalmente, o campo de operação deve ser iniciado na coluna 12.

16ª até 71ª coluna

Se existir campo em branco antes da 16ª coluna pode ser utilizado.

4. Campo de Parâmetros:

O campo de parâmetros começa pelo menos uma posição após o final do campo de operação e pode se estender até a coluna 71:

Dentro do campo de parâmetros, temos que codificar um ou mais parâmetros que fornecem informações que influenciam o modo como a instrução é processada.

Quando um campo de parâmetros for preenchido, se ele consiste em mais de um parâmetro, os parâmetros individuais devem ser separados por vírgulas.

O z/OS pressupõe que o campo de parâmetros esteja completo quando encontrar um espaço, portanto, não inclua um espaço por acidente. Se você precisar incluir um espaço como parte de um valor de parâmetro, o valor do parâmetro deverá ser colocado entre apóstrofes.

```
VIEW      KC02746.WORKBOOK.JCL(JCLEXEC2) - 01.14      Columns 00001 00072
Command ==>      Scroll ==> CSR
=COLS> -----1-----2-----3-----4-----5-----6-----7-----
***** Top of Data *****
000001 //CATPGM1 JOB ('00'),TSO.&SYSUID,
000002 //          REGION=OM,NOTIFY=&SYSUID,
000003 //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004 //*****
000005 //JOBLIB DD DSN=KC02746.LIB.USERLOAD,DISP=SHR
000006 //STEP01 EXEC PGM=COBOL021
000007 //* ARQUIVO DE ENTRADA *\\
000008 //INPUTFL DD DSN=KC02746.VAGNER.FB80QSAM,DISP=SHR
000009 //SYSTSPRT DD SYSOUT=*
000010 //SYSPRINT DD SYSOUT=*
000011 //CEEDUMP DD SYSOUT=*
000012 //SYSUDUMP DD SYSOUT=*
000013 //SYSOUT DD SYSOUT=*
000014 //REPORT DD SYSOUT=*
```

Existem dois tipos de parâmetros: Posicional e Palavra-chave

5. Campo de comentários:

Seguindo o campo de parâmetros, você pode codificar breves comentários no campo de comentários.

O campo de comentários começa na posição após o espaço que marca o final do campo de parâmetros e termina na coluna 71.

O z/OS ignora o que está codificado aqui, então você pode registrar os comentários que desejar.

Você também pode usar uma linha inteira como comentário.

Para isso, você precisa codificar `//*` começando na coluna um do registro JCL.

```
VIEW      KC02746.WORKBOOK.JCL(JCLEXEC2) - 01.14      Columns 00001 00072
Command ==>      Scroll ==> CSR
=COLS>  ----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
*****  ***** Top of Data *****
000001  //CATPGM1  JOB ('00'),TSO.&SYSUID,
000002  //          REGION=0M,NOTIFY=&SYSUID,
000003  //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004  //-----*-----
000005  //*  aponta para a biblioteca de programa  -*
000006  //*-----*-----
000007  //JOBLIB  DD  DSN=KC02746.LIB.USERLOAD,DISP=SHR
000008  //*-----*-----
000009  //*  exectuta o programa cobol021 teste de comandos em classe  *
000010  //*-----*-----
000011  //STEP01  EXEC PGM=COBOL021
000012  //*      ARQUIVO DE ENTRADA  *\\
```

Ou acrescentando espaços no final do comando e comentário

```
VIEW      KC02746.WORKBOOK.JCL(JCLEXEC2) - 01.14      Columns 00001 00072
Command ==>      Scroll ==> CSR
=COLS>  ----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
*****  ***** Top of Data *****
000001  //CATPGM1  JOB ('00'),TSO.&SYSUID,          --> JOB ID
000002  //          REGION=0M,NOTIFY=&SYSUID,
000003  //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004  //-----*-----
000005  //*  aponta para a biblioteca de programa  -*
000006  //*-----*-----
000007  //JOBLIB  DD  DSN=KC02746.LIB.USERLOAD,DISP=SHR
000008  //*-----*-----
000009  //*  exectuta o programa cobol021 teste de comandos em classe  *
000010  //*-----*-----
000011  //STEP01  EXEC PGM=COBOL021          --> exec statement
000012  //*      ARQUIVO DE ENTRADA  *\\
000013  //INPUTFL  DD  DSN=KC02746.VAGNER.FB80QSAM,DISP=SHR  --> Input file
000014  //SYSTSPRT DD  SYSOUT=*
```

Como continuar as instruções JCL?

Quando não for possível ajustar todos os parâmetros em uma única linha de 80 caracteres, a instrução pode ser continuada em uma ou mais linhas adicionais seguindo as etapas abaixo:

- Quebre o campo de parâmetro após a vírgula que segue um parâmetro ou subparâmetro.
- Barras de código nas colunas 1 e 2 da linha seguinte.
- Codifique o próximo parâmetro começando em qualquer lugar nas 4ª a 16ª coluna.

```
=COLS>  ----+-----1-----+-----2-----+-----3-----+-----4-----
*****  ***** Top of Data *****
000001  //CATPGM1  JOB ('00'),TSO.&SYSUID,
000002  //          REGION=0M,NOTIFY=&SYSUID,
000003  //          MSGCLASS=T,MSGLEVEL=1,CLASS=A
000004  //-----*-----
```

Parâmetros do JCL

Cartão: JOB

//JOBNAME JOB PARÂMETROS

ACCOUNT

Serve para identificar o Sistema e o Centro de Custo do executante do JOB, posteriormente este valor é rateado entre as diversas gerencias para cobrir o custo de processamento no Mainframe.

REGION

Aloca espaço em memória para todos os STEPs presente no JOB/PROC, este valor seria o default, caso um step necessite de mais, será alocado individualmente. O espaço pode ser em Kilobytes ou Megabytes a depender do consumo.

CLASS

No momento em que o JES2 é configurado, são atribuídas diversas classes cada uma com uma determinada prioridade que irá avançar o JOB nas filas da execução.

MSGLEVEL

Estabelece o detalhamento das informações do JOB para Output no JES2, sua sintaxe é MSGLEVEL=(X,Y). Onde:

X = declarações

Indica quais instruções de controle de tarefa o sistema deve imprimir na parte de imagens de instrução da saída JCL. Este subparâmetro é um dos seguintes números:

0

O sistema imprime a instrução JOB e todos os comentários e instruções JECL até a primeira instrução EXEC.

1

O sistema imprime todas as instruções JCL, instruções de controle JES2 ou JES3, as instruções de procedimento e mensagens IEF653I, que fornecem os valores atribuídos aos parâmetros simbólicos nas instruções de procedimento.

2

O sistema imprime apenas instruções JCL e instruções de controle JES2 ou JES3.

Y = mensagens

Indica quais mensagens o sistema deve imprimir na parte de mensagens do sistema da saída JCL. Este subparâmetro é um dos seguintes números:

0

O sistema imprime apenas mensagens JCL. Ele imprime mensagens JES e do operador somente se o trabalho for encerrado de forma anormal e imprime mensagens SMS somente se o SMS falhar no trabalho.

1

O sistema imprime mensagens JCL, JES, operadora e SMS.

TIME

Como o custo de processamento é elevado, para prevenir loops e consumo excessivo de CPU, o parametro TIME estabelece um limite de tempo para cada STEP do JOB, variando de 1 minuto até 1440 minutos (24 horas) ou segundos de 1 a 59.

TYPRUN

Parametro de fila paralela, com este comando podemos forçar que um determinado JOB fique na fila do JES2, aguardando uma autorização manual via Operações, ou também validar os comandos, enfileirando o JOB na primeira fila do JES2, que validará erros de sintaxe.

HOLD

Enfileira o JOB e aguarda liberação manual para Execução no JES2

SCAN

Solicita a análise e validação sintática do JOB, ideal para debugar erros após a criação do JCL/PROC.

LINES

Define a quantidade máxima de linhas que serão impressas durante a execução do JOB.

Cartão: JOBLIB

```
//JOBLIB DD DSN=Dataset
```

Normalmente num processamento de dados, o JCL/PROC costuma acessar inúmeros DATASETs de Bibliotecas, também chamados de Libraries, ao incluir este cartão DD, auxiliaremos a localizações dos LOADLIBs onde estão armazenados os Executáveis.

É o segundo cartão do JCL, devendo estar junto ao Cartão JOB.

Cartão: EXEC

```
//STEPNAME EXEC PARÂMETROS
```

Neste STEP iremos informar o programa Executavel e seus diversos parametros de acesso, podendo informar arquivos de Input/Output, Sysin e Sysout, Planos de acesso e parms.

Um JCL pode conter até 255 cartões EXECs por JOB, para alocar mais que o limite é necessário criar uma PROC, que irá combinar inumeros JCLs num único JOB, com a possibilidade de utilizar parametrização via Variáveis.

PGM

Recebe o nome do programa que será executado no STEP, pode receber parâmetros opcionais tais como:

PARM

Parâmetros para serem utilizados no processamento.

PROC

Recebe o nome da PROCNAME a ser executada.

COND

Parâmetro que condiciona a execução do STEP a um determinado Return-Code do STEP anterior, permitindo ignorar um STEP e validando a conclusão do STEP anterior.

REGION

Aloca espaço em memória para o STEP corrente, substituindo os valores informados no cartão JOB. O espaço pode ser em Kilobytes ou Megabytes a depender do consumo.

Sub-cartão: STEPLIB

//STEPLIB DD DSN=Dataset

Um sub-cartão que define apenas para o STEP em questão, um dataset com Librarie de LOADLIB a sere executado.

É um sub-cartão associado ao cartão EXEC, devendo ser o cartão a seguir.

Sub-cartão: DD

//DDNAME **DD** parâmetros

Neste sub-cartão iremos associar os datasets de Input/Ouput, um por cada arquivo.

DSN

DSName é o nome do arquivo a ser utilizado no cartão EXEC

DISP

Indica o que deve ser feito com o arquivo informado no DSN. Sendo que sua sintaxe é

DISP=SHR, quando for leitura/input

DISP=(XXX,YYY,ZZZ), quando for gravação/output

Onde:

XXX = NEW – novo arquivos

OLD – regrava arquivo existente

MOD – regrava, adicionando novos registros

YYY (fim normal) e ZZZ (abend) = CATLG – cataloga o arquivo independente do resultado do processamento

PASS – libera o arquivo para o próximo passo

KEEP – Mantem os arquivos

UNCATLG – Descataloga o arquivo do Catalogo

DELETE – Exclui o arquivo do disco

UNIT

Especifica a unidade de gravação, onde o DATASET será gravado. Podendo indicar um dispositivo específico ou um certo tipo ou grupo de dispositivos.

TAPE

Indica fita magnetica

DASD

Indica disco rigido

SPACE

Especifica o espaço a ser alocado e utilizado durante o processamento, tendo como subparametros o espaço inicial, o incremento e a quantidade de blocos.

SPACE=(Tamanho do bloco, (XXX, YYY, ZZZ), comando 1, comando 2, comando 3)

Tamanho do bloco TRK – Trilhas, CYL – Cilindros

XXX – quantidade primaria de espaço

YYY – quantidade secundaria de espaço

ZZZ – quantidade de blocos de 256 bytes

Comandos

RLSE – Liberar espaço alocado e não usado.

CONTIG – Obriga que o espaço seja continuo

ROUND – Arredonda para o próximo inteiro o numero de cilindros.

DCB

Informa dados físicos do arquivo, necessários para catalogar e alocá-los no sistema.

BLKSIZE

Define o tamanho da blocagem (quantidade x tamanho do registro), calculado automaticamente quando não informado.

LRECL

Tamanho em bytes do registros

RECFM

XYZ - Formato do arquivo, onde

X – Tipo de registro

F – registro de tamanho Fixo

V – registro de tamanho Variável (adiciona dois bytes ao LRECL devido ao varchar)

U – registro de tamanho Indefinido

Y - Blocagem

B – registro Blocado (opcional)

Z – Asa para impressão

A – registro com o primeiro byte indicando os saltos da impressora

DSORG

Define o tipo de DATASET a ser criado e sua organização no Catalogo do Sistema.

DA

Acesso direto em disco

PO

Arquivo de diretório, ou seja, particionado

PS

Arquivo sequencial

Subcartão - DD *

Cartão coringa, onde os dados são armazenados no JCL para Input e no SPOOL do JES2 para Output

Conhecidos como Input-Output Stream

Atributo Dummy

Coringa utilizado no subcartão DD para tratar arquivo com estado Nulo, serve tanto para Dataset de Input como Output.

//SYSIN DD *

Por convenção, as pessoas costumam usar uma instrução SYSIN DD para iniciar um conjunto de dados in-stream.

//SYSOUT DD *

//SYSUDUMP DD SYSOUT=*

//SYSABEND DD SYSOUT=*

//SYSMDUMP DD SYSOUT=*

//CEEDUMP DD SYSOUT=*

São usadas para vários tipos de dumps de memória gerados quando um programa é encerrado de forma anormal.

//SYSABOUT DD SYSOUT=*

O parâmetro SYSOUT também:

- Atribui esse conjunto de dados sysout a uma classe de saída. Os atributos de cada classe de saída são definidos durante a inicialização do JES.

- Opcionalmente, solicita que um gravador externo processe o conjunto de dados sysout em vez do JES. Um gravador externo é um programa escrito pela IBM ou por instalação.
- Identifica opcionalmente os formulários nos quais o conjunto de dados será impresso ou perfurado.
- Opcionalmente, refere-se a uma instrução JES2 /*OUTPUT para processar parâmetros.

JCL - UTILITARIO

São programas pós-escritos pela IBM e disponibilizados no TSO para auxiliar tarefas de desenvolvimento e operações, são amplamente utilizados em mainframe para atender aos requisitos do dia-a-dia, organizando e mantendo dados.

Tem inúmeras funções de produtividade, economizando tempo de codificação. Eles são usados para reorganizar, alterar ou comparar dados no conjunto de dados ou nível de registro.

Esses utilitários permitem manipular conjuntos de dados, que são fornecidos como entrada para os programas.

IEBCOPY

É usado para copiar um ou mais membros de um conjunto de dados existente para um conjunto de dados PDS novo ou existente, usado para compactar PDS, carregar PDS para TAPE e descarregar de TAPE para DISK.

IEBGNER

É usado para copiar ou imprimir conjuntos de dados sequenciais. Também usado para copiar qualquer arquivo TAPE para DISK ou DISK para TAPE.

IEHLIST

É um utilitário do sistema usado para listar entradas no diretório de um ou mais PDS, entradas no catálogo ou listar o conteúdo de um Índice de Volume (VTOC).

IEHMOVE

é usado para mover um conjunto de dados de um volume para outro.,

IEBCOMPR

É usado para comparar o conteúdo de dois conjuntos de dados PS ou PDS.

IEBEDIT

é usado para criar um JCL contendo uma seleção de etapas do trabalho. Essas etapas da tarefa podem ser inseridas no fluxo de tarefas para processamento.

IEHPROGM

É usado para excluir ou renomear conjuntos de dados ou membros e para catalogar ou descatalogar conjuntos de dados não VSAM.

DFSORT & SYNCSORT

É um poderoso utilitário IBM usado para copiar, classificar ou mesclar conjuntos de dados. A instrução SYSIN DD é usada para especificar as condições. É usado para alcançar muitas funcionalidades:

1. COPIE todos os registros de SORTIN para SORTOUT.

2. Pule primeiro 100 registros e, em seguida, copie 20 registros.
3. CLASSIFIQUE o arquivo de entrada na ordem da posição do campo especificada.
4. INCLUIR ou OMIT registro no arquivo de entrada com base na condição.
5. MERGE arquivos de entrada na ordem da posição de campo especificada.
6. JOIN dois ou mais arquivos de entrada com base no JOIN KEY especificado no arquivo de entrada.
7. Para remover duplicatas
8. SOMA os valores para registros da mesma chave.
9. Adicione o número de sequência ao arquivo de saída.
10. Reestruture o arquivo de entrada antes de alimentar a classificação.
11. Para criar vários arquivos a partir de um único arquivo de entrada.
12. Reestruture o arquivo classificado antes de gravar.

ICETOOL

é um wrapper em torno do DFSORT, é outro utilitário IBM usado para realizar uma variedade de operações em conjuntos de dados. As operações de arquivo são especificadas na instrução TOOLIN DD.

13. copie registros únicos, selecionados e duplicados.
14. Para obter estatísticas (máximo, médio e total) de um campo numérico.
15. Conte valores únicos em um campo.

IDCAMS

Gera e modifica conjuntos de dados VSAM e Grupos de Dados de Geração (GDG).

Os Comandos que realizam tarefas IDCAMS incluem: DEFINE, PRINT, LISTCAT, REPRO, ALTER, VERIFY e DELETE.

Ele cria arquivos com controle de versão para GDGs em uma base cronológica. Ele lista os atributos do conjunto de dados VSAM e exclui conjuntos de dados indesejados.