//TEDBDF41

set ENV='PROD' SET ENVB='CEND'

SET DATALIB= 'TEDWO.EDW.QA.J.DATA'

SET DATALIB1=' TEOWO, LIDNZ22.JUN2021.DATA.CMS.PROD'

SET DATALIB1='TEDWO.LIDNZ2Z.JUN2021.DATA.CMS'

SET DATALIB1='TEDWO.LIDNZ2Z.JUN2021.DATA.CMS'

SET DATALIB1='ENDEVOR.PROD.STAGE2.DATA'

SET DATALIB2='TEDW0.LIDBNX5.DATA'

EXPLAIN IT IN MAINFRAMES ONE BY ONE STEP IN DETAIL

Sure, let's go through these step by step and explain what each of these SET commands does in the context of a mainframe environment:

1. **SET ENV='PROD'**:
   * **Explanation**: This command sets an environment variable named ENV with the value PROD. This is typically used to indicate that the current environment is the production environment.
2. **SET ENVB='CEND'**:
   * **Explanation**: Similar to the previous command, this sets an environment variable named ENVB with the value CEND. This could be used to specify a particular code or configuration for the environment.
3. **SET DATALIB='TEDWO.EDW.QA.J.DATA'**:
   * **Explanation**: This sets an environment variable DATALIB to the dataset name TEDWO.EDW.QA.J.DATA. This dataset could be used for Quality Assurance (QA) purposes within the mainframe environment.
4. **SET DATALIB1='TEOWO, LIDNZ22.JUN2021.DATA.CMS.PROD'**:
   * **Explanation**: This sets an environment variable DATALIB1 to the dataset name TEOWO, LIDNZ22.JUN2021.DATA.CMS.PROD. This could be a dataset used for production with specific data for June 2021.
5. **SET DATALIB1='TEDWO.LIDNZ2Z.JUN2021.DATA.CMS'**:
   * **Explanation**: This overwrites the previous DATALIB1 variable, setting it to TEDWO.LIDNZ2Z.JUN2021.DATA.CMS. It appears to be another dataset related to the same period.
6. **SET DATALIB1='TEDWO.LIDNZ2Z.JUN2021.DATA.CMS'**:
   * **Explanation**: This is identical to the previous command and does not change anything further. The value of DATALIB1 remains TEDWO.LIDNZ2Z.JUN2021.DATA.CMS.
7. **SET DATALIB1='ENDEVOR.PROD.STAGE2.DATA'**:
   * **Explanation**: This again overwrites DATALIB1, setting it to ENDEVOR.PROD.STAGE2.DATA. This dataset name suggests it is associated with the ENDEVOR software configuration management (SCM) tool, specifically in the production environment at Stage 2.
8. **SET DATALIB2='TEDW0.LIDBNX5.DATA'**:
   * **Explanation**: This sets another environment variable DATALIB2 to the dataset TEDW0.LIDBNX5.DATA. This could be another dataset used within the environment, possibly for a different purpose or project.

These SET commands are used to configure environment variables that point to various datasets used in different stages or environments of your mainframe operations. Each dataset may contain different types of data, such as QA data, production data, or stage data, depending on your needs.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*============================================================\*

EDW EUAL Script Converter

convert EUAL BTEQ script to execute in QA environment

\*=============================================================\*

STEP005 EXEC SASTDBC

FROMTO DD DSN=4DATALIB. (EDWEUPO2), DISP-SHR

SCRPTIN DD DSN=4DATALIB2. (EDWFB01C),

DISP-SHR

SCRPTOUT DD DSN=&&TEMP,

DISP(, PASS, DELETE),

UNIT SYSDA, SPACE=(TRK, (10,20), RLSE),

DCB=(RECFM=FB, LRECL=80, BLKSIZE=0)

SASLIST DD SYSOUT=\*

SASLOG DD SYSOUT=\*

SYSIN DD DSN=TEDWO.EDW.QA.J.SAS(EDWCNVIS), DISP-SHR

Sure, let's break down each part of this JCL (Job Control Language) script step-by-step:

**Step-by-Step Explanation:**

1. **STEP005 EXEC SASTDBC**
   * **Explanation**: This line indicates the start of a new step in the job. The step name is STEP005 and it is executing a program or procedure named SASTDBC.
2. **FROMTO DD DSN=4DATALIB.(EDWEUPO2), DISP-SHR**
   * **Explanation**: This DD (Data Definition) statement defines a dataset. FROMTO is the DD name, DSN=4DATALIB.(EDWEUPO2) specifies the dataset name, and DISP=SHR means the dataset is being shared and not exclusively locked.
3. **SCRPTIN DD DSN=4DATALIB2.(EDWFB01C), DISP-SHR**
   * **Explanation**: This defines another dataset. SCRPTIN is the DD name, DSN=4DATALIB2.(EDWFB01C) specifies the dataset name, and DISP=SHR indicates the dataset is shared.
4. **SCRPTOUT DD DSN=&&TEMP, DISP=(,PASS,DELETE), UNIT=SYSDA, SPACE=(TRK,(10,20),RLSE), DCB=(RECFM=FB, LRECL=80, BLKSIZE=0)**
   * **Explanation**: This DD statement defines a temporary dataset.
     + SCRPTOUT is the DD name.
     + DSN=&&TEMP specifies the temporary dataset name.
     + DISP=(,PASS,DELETE) specifies the disposition - keep it for the next step (PASS) and delete after the job ends (DELETE).
     + UNIT=SYSDA specifies the storage unit.
     + SPACE=(TRK,(10,20),RLSE) allocates space in tracks.
     + DCB=(RECFM=FB, LRECL=80, BLKSIZE=0) specifies the dataset control block parameters: fixed block format (FB), logical record length of 80 bytes (LRECL=80), and block size set to the system default (BLKSIZE=0).
5. **SASLIST DD SYSOUT=**\*
   * **Explanation**: This directs the output listing to the system output. SASLIST is the DD name and SYSOUT=\* sends the output to the default system output class.
6. **SASLOG DD SYSOUT=**\*
   * **Explanation**: This directs the SAS log output to the system output. SASLOG is the DD name and SYSOUT=\* sends the log to the default system output class.
7. **SYSIN DD DSN=TEDWO.EDW.QA.J.SAS(EDWCNVIS), DISP=SHR**
   * **Explanation**: This DD statement defines the input dataset for the SYSIN DD. SYSIN is the DD name, DSN=TEDWO.EDW.QA.J.SAS(EDWCNVIS) specifies the dataset name, and DISP=SHR indicates the dataset is shared.

**Summary:**

This JCL step STEP005 executes the SASTDBC procedure, which processes the specified datasets (FROMTO, SCRPTIN, SCRPTOUT) and sends the output to the system output. Temporary datasets and space allocation are managed, and all outputs are directed to default system output classes. The main input dataset for this step is TEDWO.EDW.QA.J.SAS(EDWCNVIS).

\*-----------------------------------------------------\*

abend if SAS job step faila.

\*------------------------------------------------------\*

IF (STEP005.SASP.RC> 00) THEN

STEP005A EXEC PGM = ABEND

ENDIF

Let’s break down this JCL script segment, which is designed to abnormally end the job (abend) if the SAS job step fails:

**Detailed Explanation:**

1. **Comment Block**:

plaintext

\*-----------------------------------------------------\*

abend if SAS job step fails.

\*------------------------------------------------------\*

* + **Explanation**: These lines are comments. They provide a description of the following code, indicating that it will abend the job if the SAS step fails.

1. **IF (STEP005.SASP.RC > 00) THEN**:
   * **Explanation**: This IF statement checks the return code (RC) of the STEP005.SASP step. If the return code is greater than 0, it indicates an error or warning has occurred.
2. **STEP005A EXEC PGM=ABEND**:
   * **Explanation**: If the condition in the IF statement is true (i.e., the return code is greater than 0), this line executes the ABEND program. Executing this program causes the job to terminate abnormally, preventing further steps from running.
3. **ENDIF**:
   * **Explanation**: This line marks the end of the IF statement. It indicates that the conditional processing is complete.

**Summary:**

This code is a safeguard in your JCL script. It checks if the STEP005.SASP step ends with a return code greater than 0, which signifies an error. If such an error is detected, the script executes the ABEND program, causing the job to stop immediately, ensuring that no further steps are executed. This is a common practice to prevent cascading errors and manage job failures effectively.

\*-----------------------------------------------------\*

run EDW EUAL CMS FHIR EOB OP script

\*-----------------------------------------------------\*

STEP010 EXEC PGM=STOMAIN

SYSPRINT DD SYSOUT=\*

SYSUDOMF DO SYSOUT=(R,, 9511), FCB-DXLL, CHARS-GFC

SYSABEND DD SYSOUT=\*

SYSIN DD DSN=TEDWG.EDW.QA.J.DATA (EDWLOGNE), DISP-SHR

DD DSN=&&TEMP, DISP=(OLD, DELETE, DELETE)

**Detailed Explanation:**

1. **Comment Block**:

plaintext

\*-----------------------------------------------------\*

run EDW EUAL CMS FHIR EOB OP script

\*-----------------------------------------------------\*

* + **Explanation**: These lines are comments. They provide a description of the following code, indicating that it is running a script related to EDW (Enterprise Data Warehouse) EUAL (a type of processing or integration), CMS (Content Management System), FHIR (Fast Healthcare Interoperability Resources), and EOB (Explanation of Benefits) operations.

1. **STEP010 EXEC PGM=STOMAIN**:
   * **Explanation**: This line defines a new step named STEP010 that executes a program or procedure called STOMAIN.
2. **SYSPRINT DD SYSOUT=**\*:
   * **Explanation**: This DD (Data Definition) statement defines the output for system print messages. SYSPRINT is the DD name, and SYSOUT=\* sends the output to the default system output class.
3. **SYSUDOMF DO SYSOUT=(R,, 9511), FCB-DXLL, CHARS-GFC**:
   * **Explanation**: This defines the output for a dataset.
     + SYSUDOMF DO is the DD name.
     + SYSOUT=(R,,9511) directs the output to a specific output class (R) and a specific destination (9511).
     + FCB=DXLL specifies a Form Control Buffer for print control.
     + CHARS=GFC specifies character translation for printing.
4. **SYSABEND DD SYSOUT=**\*:
   * **Explanation**: This DD statement defines the output for system abend (abnormal end) messages. SYSABEND is the DD name, and SYSOUT=\* sends the output to the default system output class.
5. **SYSIN DD DSN=TEDWG.EDW.QA.J.DATA(EDWLOGNE), DISP=SHR**:
   * **Explanation**: This defines the input dataset for the SYSIN DD.
     + SYSIN is the DD name.
     + DSN=TEDWG.EDW.QA.J.DATA(EDWLOGNE) specifies the dataset name.
     + DISP=SHR indicates that the dataset is shared and not exclusively locked.
6. **DD DSN=&&TEMP, DISP=(OLD,DELETE,DELETE)**:
   * **Explanation**: This line adds another input dataset for SYSIN.
     + DSN=&&TEMP specifies a temporary dataset.
     + DISP=(OLD,DELETE,DELETE) means the dataset will be accessed in old status (OLD), and it will be deleted after the step and after the job completes.

**Summary:**

This JCL step STEP010 executes the STOMAIN program and defines various datasets for input and output. It includes settings for system print messages (SYSPRINT), specific output control (SYSUDOMF DO), and system abend messages (SYSABEND). The main input datasets for this step are TEDWG.EDW.QA.J.DATA(EDWLOGNE) and a temporary dataset &&TEMP.

\*-----------------------------------------------------\*

abend if teradata BTEQ fails

\*-----------------------------------------------------\*

IF (STEP010.BC> 00) THEN

STEP005A EXEC PGM=ABEND

Let's break down and explain this JCL segment, which ensures the job abends (abnormally ends) if the Teradata BTEQ step fails:

**Detailed Explanation:**

1. **Comment Block**:

plaintext

\*-----------------------------------------------------\*

abend if teradata BTEQ fails

\*-----------------------------------------------------\*

* + **Explanation**: These lines are comments that describe the purpose of the code, indicating that it will cause an abend if the Teradata BTEQ step fails.

1. **IF (STEP010.BC > 00) THEN**:
   * **Explanation**: This IF statement checks the return code (RC) of STEP010.BC. If the return code is greater than 0, it indicates that the BTEQ step encountered an error.
2. **STEP005A EXEC PGM=ABEND**:
   * **Explanation**: If the condition in the IF statement is met (i.e., the return code is greater than 0), this line executes the ABEND program, which forces the job to terminate abnormally.
3. **ENDIF**:
   * **Explanation**: This marks the end of the IF statement, indicating that the conditional logic is complete.

**Summary:**

This JCL segment is designed to handle errors that occur during the Teradata BTEQ step (STEP010). If the BTEQ step results in a return code greater than 0, indicating an error, the job will execute the ABEND program to stop further execution and terminate the job. This ensures that any issues are caught and the job is halted to prevent further problems.

\*-----------------------------------------------------\*

run EDW EUAL CMS FHIR EOB OP script

\*-----------------------------------------------------\*

STEP010 EXEC PGM=BTOMAIN

SYSPRINT DD SYSOUT='

SYSUDUMP DO SYSOUT =(R..9511), FCB-DXLL, CHARS-GFC

SYSABEND DD SYSOUT='

//SYSIN DD DSN=TEDWO.EDW.QA.J.DATA (EDWLOGNE), DISP=SHR

DO DSN=&&TEMP, DISP (OLD, DELETE, DELETE)

Sure! Let's break down and explain this JCL segment step by step:

**Detailed Explanation:**

1. **Comment Block**:

plaintext

\*-----------------------------------------------------\*

run EDW EUAL CMS FHIR EOB OP script

\*-----------------------------------------------------\*

* + **Explanation**: These lines are comments that describe the purpose of the following code, indicating that it will run a script related to EDW (Enterprise Data Warehouse), EUAL (presumably a type of processing or integration), CMS (Content Management System), FHIR (Fast Healthcare Interoperability Resources), and EOB (Explanation of Benefits) operations.

1. **STEP010 EXEC PGM=BTOMAIN**:
   * **Explanation**: This line defines a new step named STEP010 that executes a program or procedure called BTOMAIN.
2. **SYSPRINT DD SYSOUT='**:
   * **Explanation**: This DD (Data Definition) statement is meant to define the output for system print messages. SYSPRINT is the DD name, but it appears to be incomplete as it lacks the output class definition (e.g., SYSOUT=\*).
3. **SYSUDUMP DO SYSOUT=(R..9511), FCB=DXLL, CHARS=GFC**:
   * **Explanation**: This defines the output for a dataset.
     + SYSUDUMP DO is the DD name.
     + SYSOUT=(R..9511) directs the output to a specific output class (R) and a specific destination (9511).
     + FCB=DXLL specifies a Form Control Buffer for print control.
     + CHARS=GFC specifies character translation for printing.
4. **SYSABEND DD SYSOUT='**:
   * **Explanation**: This DD statement is meant to define the output for system abend (abnormal end) messages. SYSABEND is the DD name, but it appears to be incomplete as it lacks the output class definition (e.g., SYSOUT=\*).
5. **SYSIN DD DSN=TEDWO.EDW.QA.J.DATA(EDWLOGNE), DISP=SHR**:
   * **Explanation**: This defines the input dataset for the SYSIN DD.
     + SYSIN is the DD name.
     + DSN=TEDWO.EDW.QA.J.DATA(EDWLOGNE) specifies the dataset name.
     + DISP=SHR indicates that the dataset is shared and not exclusively locked.
6. **DO DSN=&&TEMP, DISP=(OLD,DELETE,DELETE)**:
   * **Explanation**: This line adds another input dataset for SYSIN.
     + DO is likely a typo or misprint, it should probably be DD.
     + DSN=&&TEMP specifies a temporary dataset.
     + DISP=(OLD,DELETE,DELETE) means the dataset will be accessed in old status (OLD), and it will be deleted after the step and after the job completes.

**Summary:**

This JCL step STEP010 is designed to execute the BTOMAIN program and manage various datasets for input and output. It includes settings for system print messages (SYSPRINT), specific output control (SYSUDUMP), and system abend messages (SYSABEND). The main input datasets for this step are TEDWO.EDW.QA.J.DATA(EDWLOGNE) and a temporary dataset &&TEMP. However, there seem to be a couple of incomplete or typo issues in the provided script, which should be addressed to ensure proper execution.

\*-----------------------------------------------------\*

abend if teradata BTEQ fails

\*-----------------------------------------------------\*

IF (STEP010.RC>00) THEN

STEP010A EXEC PGM=ABEND

ENDIF

Let's break down this JCL segment step by step to understand how it ensures the job abends (abnormally ends) if the Teradata BTEQ step fails:

**Detailed Explanation:**

1. **Comment Block**:

plaintext

\*-----------------------------------------------------\*

abend if teradata BTEQ fails

\*-----------------------------------------------------\*

* + **Explanation**: These lines are comments. They provide a description of the following code, indicating that it will cause an abend if the Teradata BTEQ step fails.

1. **IF (STEP010.RC>00) THEN**:
   * **Explanation**: This IF statement checks the return code (RC) of the STEP010 step. If the return code is greater than 0, it indicates that the BTEQ step encountered an error.
2. **STEP010A EXEC PGM=ABEND**:
   * **Explanation**: If the condition in the IF statement is met (i.e., the return code is greater than 0), this line executes the ABEND program, which forces the job to terminate abnormally.
3. **ENDIF**:
   * **Explanation**: This line marks the end of the IF statement, indicating that the conditional logic is complete.

**Summary:**

This JCL segment is designed to handle errors that occur during the Teradata BTEQ step (STEP010). If the BTEQ step results in a return code greater than 0, indicating an error, the job will execute the ABEND program to stop further execution and terminate the job. This ensures that any issues are caught and the job is halted to prevent further problems.