#### Unclassified

# **Internal Design**

#### \*Business Area Custom Build

#### Library

# STAG Language/Type Rexx

\*Function CBDMCMD -> Create CB DVD Order Media (CBDMCMD)

\*Process Create CB DVD Order Media (CBDMCMD)

\*Project(s) 11076 Productive Development Miscellaneous Tasks

History of all Projects: 11076 Productive Development Miscellaneous Tasks,12465 CB DOS Changes and Testing/Deployment of CDWRITE/UTD/Windows 7 client software for DVD towers communication conversion from CTC to TCP/IP,12419 Take over support of Action Base Code and Production Center SKELs,12439 Test and Deploy updated CDWRITE module in support of RCQ00225875,12296 Provide Mfg Support For z/OS Version 1 Rel 13,12085 Provide Mfg Support For z/OS Version 1 Rel 12

Created: 19-05-2010 12:17:59 PM by CN=Carlos Eduardo da Silva/OU=Brazil/OU=Contr/O=IBM

Last Modified: 28-11-2018 3:59:47 PM by Hector Tello1/Mexico/IBM

·

----

# ▼ Module Inputs/Parameters Parameter Inputs

# **Change History:**

- New Internal Design for WI# 12085 (CES 05/13/2010)
- Changes for WI# 12296 are hilited in blue (AMS 05/05/2011).
- Changes for WI# 11076 PR# 8788 and TSRM incident IN1473094 are hilited in **dark blue** (AMS 11/10/2011)
- Changes for WI#11076 PR# 9250 are hilited in dark Green (APM 02/06/2013)
- Changes for WI#11076 PR# 9372 are hilited in dark red (AMS 06/07/2013).
- Changes for WI#12439 are hilited in light blue (AMS 16/07/2013).
- Changes for WI# 11076 PR#9595 are hilited in light brown (AMS 27/12/2013).
- Changes for WI# 12419 (Phase 1C) are hilited in light purple (AMS 10/29/2015)
- Changes for WI# 12465 RRC 158082, Portfolio Epic 1233412, and User Story 1412722 are hilited in dark magenta (MRODOLF 02/07/2017)
- Additional changes for WI# 12465 RTC Defect 1436284 are hilited in **light green** (MRODOLF 03/01/2017) and (AMS 03/05/2017)
- Additional changes for WI# 12465 (Phase 1B) RRC 158082, Portfolio Epic 1233412, and User Story 1490368 are hilited in rouge (MRODOLF 05/25/2017)
- -Changes for WI# 11076 in support for Defect 1830554 are hilited in color Hot Pink (HTELLO 11/28/18).

#### **Database Inputs**

# CB Mfg. Order Table (VCBMORDER)

- Contains rows for CBPDO, ServerPac, and CustomPac DVD **and cartridge** orders (one row for each order processed).
- Contains rows for COD DVD orders

## File Inputs

# Create CB DVD Order Media Configuration File (CTL)

- Contains parms used by this process.

# **CB DVD Order PREMAST Image File**

- An ISO image file created by the **PREMAST program** called by the **Create CB DVD Order PREMAST Image File (CBDMPPR) process** 

## CB Mfg. Products File (CBORDPR):

- Contains records (Type '1 ' through Type '4 ') for CBPDO, ServerPac, and CustomPac DVD , and cartridge orders,
- Contains records (Type '1 ' through Type '4 ') for COD DVD orders.

# CB IP Connected DVD Tower Details File (IPCFG)

- Contains DVD Tower record(s) and Unit records (DVD drive unit numbers) corresponding to CB IP Connected DVD Tower(s).

# CB DVD Unit Addresses File

- Contains record for a CB DVD Unit Address to be allocated, written to, and free'd for a CB DVD order.

# CB DVD Order Processing Warning Notification File (WARNTFY)

- Lotus Notes E-mail that contains information regarding a CBDMCMD processing warning that may occur.

# CB DVD Order Processing Error Notification File (ERRNTFY)

- Lotus Notes E-mail that contains information regarding a CBDMCMD processing error that may occur.

# **Other Inputs**

▼ Module Outputs/Parameters

Parameter Outputs

### **Database Outputs**

# **File Outputs**

#### CB DVD Unit Addresses File

- Contains record for a CB DVD Unit Address to be allocated, written to, and free'd for a CB DVD order.

# **Reports**

# Create CB DVD Order Media Summary Report

- Will contains lines for the:
- Number of CB Mfg. Products File processed

- Number of Type '4' records processed in the CB Mfg. Products File
- Number of Type '4' records bypassed in the CB Mfg. Products File
  - Number of DVD volumes created

### **Other Outputs**

# OMS Admin Queue ::

- Queue used for sending an **OMS Admin message**...
- The OMS SDS Logger process will log the error in the OMS Activity Log Table .
- The queue manager and queue name will be specified by the **OMS Admin Queue Name parm** (**OMSADMINQNAME**)

in the Create CB DVD Order Media Configuration File (CTL)

## **Modules Called**

- **\$CFG** - Cfg file module

- **\$LOG** - Create SDF Log entry

- \$MQI

- MQ Series Rexx Functions

- OMSBM - OMS Message Builder (for OMS Admin Message )

- CB\$PREF - Email Notification processing Function

- **\$DBICB** - RXSQL Interface Functions

- SXMLBM - Build XML message Generic Routine

- CBMDYT - TIMESTAMP function

- CDWRITE - Write CB DVD order image file

#### Called By

none

#### **Error Handling**

See Message Definitions

# **Message Definitions**

# The following SYSPRINT messages will result in an ABEND with RC=12

CBDMCMD001E: NO RECORDS TO PROCESS IN CB MFG PRODUCTS FILE

CBDMCMD002E: REXX Error <rc>in line <sigl>: <errortext(rc) ||> . Source line= <sourceline(sigl)>

CBDMCMD003E: REXX Error: Uninitialized variable condition(<D>) on line <sigl>

CBDMCMD004E: READ UNSUCCESSFUL FOR <DDName var> DD

CBDMCMD005E: ERROR TRYING TO CONNECT TO QMANAGER <Cfg.QMgr>

CBDMCMD006E: Error in CfgValidate - Invalid CFG input parameters

CBDMCMD007E: Error in CfgRead - <CfgDD> File open failed

CBDMCMD008E: NO TYPE 4 RECORDS FOUND FOR CB MFG ORDER #: <CB Mfg Order field>

CBDMCMD009E: NO ROW FOUND IN VCBMORDER FOR CB MFG ORDER#:

<VCBMORDER\_IMFGORD>

CBDMCMD010E: DB2 ERROR OCCURRED TRYING TO SELECT FROM VCBMORDER FOR CB MFG

ORDER#:<VCBMORDER IMFGORD>

CBDMCMD012E: CB\$PREF failed with RC= < Process Notif rc>

CBDMCMD013E: ERROR TRYING TO PUT MESSAGE TO OMSADMINQNAME

<Cfg.OMSADMINQName>

CBDMCMD014E: ERROR TRYING TO OPEN OMSADMINQNAME < Cfg.OMSADMINQName > CBDMCMD015E: Error occurred updating record in < CB DVDUnitAddressesFileName var>

CBDMCMD017E: Unable to locate record for <Job Name var> in file

<CB DVDUnitAddressesFileName var>

CBDMCMD018E: CBMDYT FAILED WITH RC= <MYRC>

CBDMCMD022E: DD Name <DD\_Name\_for\_CB\_DVDTowerUnitNumber\_var> and DVD Unit Address

<CB DVDUnitAddress var> was not allocated successfully

CBDMCMD026E: CDWRITE encountered an error and ended with an RC of <CD\_rc>

CBDMCMD028E: Order#: <VCBMORDER\_IESWORDER>, Line Item#:<VCBMORDER\_IESWLITEM>, CB Mfg. Order#: <VCBMORDER\_IMFGORD#>, CB Mfg. Run#: <VCBMORDER\_IMFGRUN#> has been cancelled and no DVDs will be created. The DVDCRT Action for the order should be set to a status of Cancel.

CBDMCMD029E: Unable to locate record for <CB\_DVDUnitAddress\_var> in the CB IP Connected DVD Tower Details File (IPCFG DD)

CBDMCMD030E: Connection Type field value in a CB DVD Unit Addresses File <

CB\_DVDUnitAddressesFileName\_var> record not I or C

# The following SYSPRINT messages will result in RC=04 and in a warning

CBDMCMD020W: CB DVD Unit Addresses File <CB\_DVDUnitAddressesFileName\_var> could not be allocated for Action Type <Action\_Type\_var> after <Max\_Number\_Try\_alloc> tries

## The following SYSPRINT message is only for information

CBDMCMD011I: CB DVD Unit Addresses File <DSNAME\_var> could not be allocated for Action Type <Action\_Type\_var> will try and re-allocate in

<cfg.SLEEP SECONDS> seconds

CBDMCMD016I: CB DVD Unit Addresses File <DSNAME\_var> allocated for Action Type

<Action Type var>

CBDMCMD019I: Record for DVD Unit Address < CB\_DVDUnitAddress\_var> and Action Type

<a href="#"><Action Type var> was updated successfully in CB DVD</a>

Unit Addresses File < CB DVDUnitAddresses File Name var

CBDMCMD021I: DD Name < DD Name for CB DVDTowerUnitNumber var> and DVD Unit Address

<CB DVDUnitAddress var> was allocated successfully

CBDMCMD023I: Bypassing call to CDWRITE

CBDMCMD024I: Calling CDWRITE for CB DVD Order PREMAST Image File

<CB DVDOrderPREMASTImageFileName var>

using DD Name <DD\_Name\_for\_CB\_DVDTowerUnitNumber\_var> and CB DVD Unit Address <CB\_DVDUnitAddress\_var>

CBDMCMD025I: CDWRITE processed CB DVD Order PREMAST Image File

<B DVDOrderPREMASTImageFileName var> successfully

CBDMCMD27I: CDWRITE bypassed processing CB DVD Order PREMAST Image File

<CB DVDOrderPREMASTImageFileName var>

CBDMCMD099I: CBDMCMD terminated with rc=<CBDMCMD RC>

## **Module Description**

- -This process (Create CB DVD Order Media (CBDMCMD) ) will do the following:
- The purpose of this module:
- Process the Type '1 ' and Type 4 records in the **CB Mfg. Products File** (created for CBPDO, ServerPac, and CustomPac, and COD cartridge and DVD orders)

and CBPDO, ServerPac, and CustomPac cartridge orders)

to obtain the CB Mfg. Order# field (positions 9 -14) which will then be used to obtain the order details (ESW Order#, ESW Line Item#, CB Offering Type, CB Mfg Run#, CB Offering Type)

for the order from the CB Mfg. Order Table (VCBMORDER)

- If the order is in a Cancelled Status then:
- An OMS Admin Notification MQ Series message will be created and put on the queue manager name specified by the Queue Manager name parm (QMGR) and the queue name specified by the OMS Admin Queue Name parm (OMSADMINQNAME) in the

Create CB DVD Order Media Configuration File

- The process will end with an return code of 12.
- Process the record in the **CB DVD Unit Addresses File** assigned to the job name being run to obtain the DVD unit address to be used.
- Update the record in the **CB DVD Unit Addresses File** assigned to the job name being run to indicate the job has started.
- -Perform the Process CB DVD Unit Addresses File Routine passing it the Action Type variable.
- Allocate the CB DVD Unit Address to a DD name.-with the following attributes:
- For each Type '4' record in the CB Mfg. Products File:
- If the record is for a DVD volume :
- Derive the value of the CB DVD Order PREMAST Image File Name variable

(name of the CB DVD Order PREMAST Image File )

-Call the Perform CDWRITE program If the value of the Should CDWRITE be called parm (CALL\_CDWRITE)

from the Create CB DVD Order Media Configuration File (CTL) indicates the CDWRITE program should be called.

- If the record in the CB DVD Unit Addresses File assigned to the job name indicates the CB DVD Unit Address to be allocated is for a CB CTC Connected DVD Tower:
- The DVD Unit Address will be allocated to a DD name obtained from a parm specified in the Create CB DVD Order Media Configuration File.
- If the record in the CB DVD Unit Addresses File assigned to the job name indicates the CB DVD Unit Address to be allocated is for a CB IP Connected DVD Tower:
- The DVD Unit Address will be allocated to a DD name obtained from a corresponding unit record in the CB IP Connected DVD Tower Details File.
- Call the CDWRITE program passing it parms for the name of the CB DVD Order PREMAST Image

**DD** name assigned to the **DVD** Unit Address

and VOLSER obtained from the VOLSER field in the record being processed

- Update the record in the **CB DVD Unit Addresses File** corresponding to the job name being run to indicate the job has completed.
- If the record is not for a DVD volume (could be for a cartridge volume) then:
- It will be bypassed
- -IF any processing errors or warning are encountered then:
- An **OMS Admin Notification** MQ Series message will be created and put on the queue manager name specified by the **Queue Manager name parm (QMGR)** and the queue name specified by the **OMS Admin Queue Name parm (OMSADMINQNAME)** in the **Create CB DVD Order Media Configuration File**.
- The message will be used by the OMS Process Admin Notification Msgs process to send out a Create CB DVD Order Media (CBDMCMD) process warning e-mail notification or Create CB DVD Order Media (CBDMCMD) process error e-mail notification to the email addresses in the CB Order Processing/Tracking Process Error Notification File

**Structure Chart (optional)** 

# **Module Design**

Include the following external modules

INCLUDE \$CFG /\*Read Control File Generic Routine \*/
INCLUDE \$XMLBM /\*Build XML message Generic Routine \*/
INCLUDE \$LOG /\*Log Generic Routine \*/

INCLUDE \$MQI	/* MQSeries Generic Routine		*/
INCLUDE OMSBM	/*OMS Build MetaData Generic Routine	*/	
INCLUDE \$DBICB	/*RXSQL Interface Function	*/	
INCLUDE CB\$PREF	/*Email Notification processing Function	า */	
INCLUDE CBMDYT	/*TIMESTAMP FUNCTION	*/	
INCLUDE CDWRITE /*Write	CB DVD order image file	*/	

#### Main Routine:

Description: Process the Type '1 and Type '4' records in the CB Mfg. Products File ; Perform the Process CB DVD

Unit Addresses File Routine passing it the Action Type variable; Allocate the CB DVD Unit Address with the following attributes; For each Type '4' record that is for a DVD volume: Derive the value of the CB DVD Order PREMAST Image File Name variable (name of the CB DVD Order PREMAST Image File ), Perform CDWRITE module If the value of the Should CDWRITE be called parm (CALL\_CDWRITE) from the Create CB DVD Order Media Configuration File (CTL) indicates the CDWRITE program should be called. The DVD Unit Address will only be allocated when the CALL\_CDWRITE parm

from the Create CB DVD Order Media Configuration File (CTL) indicates the CDWRITE program should be called;

Calls routines to do the following:

- Configuration File processing (Read\_CTL\_File);
- Initialization processing (Initialize);
- Read CBORDPR file (Read File);
- Process CBORDPR file record Type '1 ' (Process\_Type\_1), obtain the order details for the

order being processed from the CB Mfg. Order Table (Select\_VCBMORDER) and verify if the order is in a valid status to process.

- Allocate CB DVD Unit Address File, obtain the connection type to be used and update the record with the

current status of 'UPDATE STARTED' (Process CBDVDUnitAddressesFile)

- If Configuration File cfg.CALL CDWRITE = 'Y':

If the Connection type = 'C'

- Determine and allocate a CB DVD Unit Address (Allocate\_CBDVDUnitAddress) If the Connection type = 'I'
- Determine the CB DVD Unit Record to be allocated (Process CBIPConnectedDVDTowerDetailsFile)
- Allocate the CB DVD Unit Record (Allocate CBDVDUnitAddress)
- Process each CBORDPR file record Type '4' for a DVD media type and send the correspondent

CB DVD IMG file to produce a DVD media via CDWRITE program (Process Type 4)

- Updated the CB DVD Unit Address File record with the status of 'UPDATE COMPLETED',

create the Summary Report and end the processing (Program\_End).

Inputs: noneCBORDPR\_REC stem, cfg.CALL\_CDWRITE, CONNTYPE\_var, cfg.CB\_DVD\_UNIT\_DD\_NAME

Outputs: CBDMCMD\_RC.IF error, return\_code of 12, error email notification-Action\_Type\_var, CB\_DVDUnitAddress\_var,DD\_Name\_for\_CB\_DVDTowerUnitNumber\_var', Msg#, MsgCode,

signal on syntax signal on novalue numeric digits 15

```
parse source . . execName . /* Get exec name */
Call LogInit execName
Call Read_CTL_File
Interpret "trace" cfg.traceOption
Call Initialize
Call Read File with arguments 'CBORDPR' for DDName, 'CBORDPR' REC.' for StemName and 'Y'
for Free parm.
IF CBORDPR REC.0 > 0 THEN
DO
   DO Ind = 1 to CBORDPR REC.0
       SET Order Type = SUBSTR(CBORDPR REC.ind,1,2)
       Select
          When Order Type = '1 'THEN
          DO
              SET Type 1 Count = Type 1 Count + 1
              SET CBORDPR Tp1 REC.Type 1 Count = CBORDPR REC.ind
          END
          When Order Type = '4 ' THEN
          DO
              SET Type_4_Count = Type 4 Count + 1
              SET CBORDPR Tp4 REC.Type 4 Count = CBORDPR REC.ind
          OTHERWISE NOP
       END
   END
   Call Process_Type_1
   Call Process_CBDVDUnitAddressesFile with argument 'Action_Type_var' for Action Type
   IF cfg.CALL CDWRITE = 'Y' THEN
   DO
   IF CONNTYPE var = 'C' THEN
   DO
   SET All UA rc = Allocate CBDVDUnitAddress with arguments
   'cfg.CB_DVD_UNIT_DD_NAME' for DDname and 'CB_DVDUnitAddress_var' for DSName
   SET DD_Name_for_CB_DVDTowerUnitNumber_var = cfg.CB_DVD_UNIT_DD_NAME
   END
   ELSE
   DO
   CALL Process_CBIPConnectedDVDTowerDetailsFile with argument CB DVDUnitAddress var
   SET CB DVDUnitAddress var = 'VIO'
   SET All UA rc = Allocate CBDVDUnitAddress with arguments
   'DD Name for CB DVDTowerUnitNumber var' for DDname and 'CB DVDUnitAddress var' for
   DSName
   END
   IF All UA rc = 0 THEN
   SET CB DVDUnitAddressDDNameAllocated var = 'Y'
       Call Format and Send Message with arguments '21' for Msg# and 'I' for MsgCode
   Call Format_and_Send_Message with arguments '24' for Msg# and 'I' for MsgCode
       Call Process_Type_4
      "FREE FI("cfg.CB_DVD_UNIT_DD_NAME")"
```

```
END
       ELSE
       DO
       SET CB DVDUnitAddressDDNameAllocated var = 'N'
           Call Format and Send Message with arguments '22' for Msg# and 'E' for MsgCode
           SET CBDMCMD RC = 12
       END
END
       IF CB DVDUnitAddressDDNameAllocated var = 'Y' OR cfg.CALL CDWRITE = 'N' THEN
       Call Process_Type_4
       END
   END
   ELSE
       Call Format_and_Send_Message with arguments '01' for Msg# and 'E' for MsgCode
   Call Program End
Description: This procedure send a information message and end the process.
- Disconnect from MQSeries via the MqiDisconnect routine
- Call the DBIFinal routine to end the REXX DB2 interface
- Call the Format and Send Message routine to write out a final informational message
with arguments '99' for Msg# and 'I' for MsgCode
- Exit using the CBDMCMD RC value for the return code
Called by : all routines.
Inputs: CBDMCMD RC Msg# MsgCode
Outputs: CBDMCMD_RC
exitPt:
   SET Myrc = MqiDisconnect()
   Call DBIFinal
   Call Format and Send Message with arguments '99' for Msg# and 'I' for MsgCode
   Exit CBDMCMD RC
Rexx Error Handling Labels, will be called automatic when a Error Handling occurs.
SYNTAX:
              /* Catch REXX syntax errors */
   SET lin var = sigl
   Call Format and Send Message with arguments '02' for Msg# and 'E' for MsgCode
NOVALUE:
                     /* Catch not initialized variable error */
   SET lin var = sigl
   Call Format and Send Message with arguments '03' for Msg# and 'E' for MsgCode
```

### Read CTL File Routine:

Description: This procedure gets and validates program parameters contained in the CTL DD file.

It uses two \$CFG included routines, CfgRead which opens the file DD specied in the cfgDD

arg then reads all the parameters in the configuration file into an array and CfgValidate.

arg then reads all the parameters in the configuration file into an array and CfgValidate which validates all keywords in the array against the parameters specified in the ctlKeys arg.

Called by : Main Routine.

```
if system error, CBDMCMD RC of 12 & SYSPRINT log message
   SET CfqDD = "CTL"
   SET CfgKeys = "QMGR JOB NAME PREFIX CB DVD UNIT DD NAME".
                "CALL_CDWRITE CDWRITE_PGM_NAME_LOAD CB_DVD_ORD_FILE_HLQ",
                "CB_DVD_ORD_PREMAST_FILE_LLQ CB_DVD_ORD_UA_FILE_LLQ",
                "SLEEP_SECONDS MAX_TRIES_TO_ALLOC DVD_MEDIA_TYPES
                OMSADMINQNAME".
                "OMS CODEPAGE EMAIL ID CHAR",
                "DB2SYS DBATTEMPTS DBWAIT REM DB2 CONN",
                "PLAN NAME RXSQL TRACEOPTION",
                "CANC STATUS""
   SET Myrc = CfgRead(CfgDD)
   If Myrc = 0 THEN
   DO
      SET Myrc = <u>CfgValidate</u>(CfgKeys)
      If Myrc <> 0 THEN
          Call Format_and_Send_Message with arguments '06' for Msg# and 'E' for MsgCode
   END
   ELSE
      Call Format and Send Message with arguments '07' for Msg# and 'E' for MsgCode
End Read_CTL_File Routine;
Initialization Routine:
Description: These includes are: MqiInit to MQ initialization, MqiConnectToQMqr connect to the MQ Mqr
listed in the
```

QMGR parameter in the CTL DD file and DBIINIT to connect to DB2 system. (Note: ERRNTFY DD email file is open/read later by the CB\$PREF included routine.)

Called by: Main routine.

Inputs: cfg.stem

Outputs:

CBDMCMD RC.IF error, return code of 12 & SYSPRINT log message

```
SET CBDMCMD RC = 0
   SET CBPDO order = 'N'
   SET ServerPac order = 'N'
   SET COD order = 'N'
   SET Total VOL Created = 0
   SET Type 1 Count = 0
   SET Type_4_Count = 0
   SET Type_4_Processed = 0
   SET Type 4 Bypassed = 0
SET Found DVD Media Type = 'N'
   SET CB_DVDUnitAddressesFileName_var = cfg.CB_DVD_ORD_FILE_HLQ || '.' ||
   cfa.CB DVD ORD UA FILE LLQ
SET CB DVDUnitAddressDDNameAllocated var = 'N'
   SET CONNTYPE var = "
   SET DD_Name_for_CB_DVDTowerUnitNumber_var = "
   SET CBDVDUN var= "
```

```
SET MATCH CBDVDUN = 'N'
   SET CB DVDUnitAddress parm= "
   SET RECTYPE var= "
   SET Final CB DVDUnitAddressesFileUpdate var = 'Y'
   CALL DBIINIT CFG.DB2SYS, CFG.DBATTEMPTS, CFG.DBWAIT, 8
   SET Myrc = MqiInit('8')
   IF myrc = 0 THEN
       SET myrc = MqiConnectToQMgr(Cfg.QMgr)
   If myrc <> 0 THEN
       Call Format and Send Message with arguments '05' for Msg# and 'E' for MsgCode
End Initialization Routine:
Read File Routine:
Description: This procedure is a common routine to read the file specified in the DDname argument and
write the file
               records to the StemName specified in the argument
Called by: Main, Process Type 4, Get Media Type Process CBIPConnectedDVDTowerDetailsFile
routines
Inputs: DDName parm StemName parm Free parm
Outputs: StemName,
               CBDMCMD RC.IF error, return code of 12, error mail notificatio & SYSPRINT log
               message
   Parse Arg DDName_parm StemName_parm Free_parm
   'EXECIO * DISKR' DDName' 0 (FINIS STEM' StemName
   IF rc <> 0 THEN
       Call Format and Send Message with arguments '04' for Msg# and 'E' for MsgCode
   IF Free Parm = 'Y' THEN
       "FREE FI("DDName")"
End Read File Routine;
Process Type 1 Routine:
Description: Process CBORDPR type 1 record.
Called by : Main Routine
Inputs: none cfg stem, Type_1_Count, CBORDPR_Tp1_REC.Type_1_Count, CBPDO_order,
ServerPac order,
COD order, VCBMORDER IMFGRUN#
Outputs: none Job_Name_var, CB_Mfg_Order_field, Action_Type_var
   DO Type 1 This Count = 1 TO Type 1 Count /* Will be just 1 record */
       SET CB Mfg Order field=SUBSTR(CBORDPR Tp1 REC.Type 1 This Count,9,6)
       Call Select VCBMORDER
   END
   SET Action_Type_var = 'UPDATE STARTED'
```

```
IF CBPDO order = 'Y' OR ServerPac order = 'Y' OR COD order = 'Y' THEN
   SET Job Name var = cfg.JOB NAME PREFIX || Substr(CB_Mfg_Order_field,2,5)
   ELSE /* CustomPac order is being processed */
   SET Job Name var = cfg.JOB NAME PREFIX || Substr(VCBMORDER IMFGRUN#,4,5)
   END
   SET CB_DVDUnitAddressesFileName_var = cfg.CB_DVD_ORD_FILE_HLQ || '.' ||
   cfg.CB DVD ORD UA FILE LLQ
End Process_Type_1 Routine;
Select VCBMORDER Routine:
Description: Obtain the order details for the order from the CB Mfg. Order Table (VCBMORDER)
Called by: Process Type 1 Routine
Inputs: CB Mfg Order field
Outputs: IESWORDLI, IESWORDER, IESWLITEM, IPROD#, IMFGRUN#, IMFGORD#, ISTATUS from
VCBMORDER
              CBDMCMD RC.IF error, return code of 12, error email notification & SYSPRINT log
              Message
   SET VCBMORDER IMFGORD = CB Mfg Order field
   SELECTSTMT = "SELECT IESWORDER, IESWLITEM, IMFGRUN#, ICBTYPE,
   IMFGORD#, ISTATUS".
   "FROM CBS.VCBMORDER",
   "WHERE IMFGORD# = ""VCBMORDER IMFGORD"""
   SET DBIISSUESQL RC = DBIISSUESQL(SELECTSTMT)
   SELECT
      WHEN (DBIISSUESQL RC = 0) THEN
      DO
          SET VCBMORDER IESWORDER = STRIP(IESWORDER.1)
          SET VCBMORDER IESWLITEM = STRIP(IESWLITEM.1)
          SET VCBMORDER IMFGRUN# = STRIP(IMFGRUN#.1)
          SET VCBMORDER ICBTYPE = STRIP(ICBTYPE.1)
          SET VCBMORDER IMFGORD# = STRIP(IMFGORD#.1)
          SET VCBMORDER ISTATUS = STRIP(ISTATUS.1)
```

```
SET DBIISSUESQL_RC = DBIISSUESQL(SELECTSTMT)

SELECT
WHEN (DBIISSUESQL_RC = 0) THEN
DO

SET VCBMORDER_IESWORDER = STRIP(IESWORDER.1)
SET VCBMORDER_IESWLITEM = STRIP(IESWLITEM.1)
SET VCBMORDER_IMFGRUN# = STRIP(IMFGRUN#.1)
SET VCBMORDER_IMFGRUN# = STRIP(IMFGRUN#.1)
SET VCBMORDER_IMFGORD# = STRIP(IMFGORD#.1)
SET VCBMORDER_ISTATUS = STRIP(ISTATUS.1)

IF VCBMORDER_ISTATUS = STRIP(ISTATUS.1)

IF VCBMORDER_ISTATUS = Cfg.CANC_STATUS THEN
Call Format_and_Send_Message with arguments '28' for Msg# and 'E' for MsgCode
ELSE
IF VCBMORDER_ICBTYPE = 'CBPDO' THEN
SET CBPDO_order = 'Y'
ELSE
IF VCBMORDER_ICBTYPE = 'SERVERPAC' THEN
SET ServerPac_order = 'Y'
ELSE
IF VCBMORDER_ICBTYPE = 'COD' THEN
SET COD_order = 'Y'
END
WHEN (DBIISSUESQL_RC = 100) THEN
Call Format_and_Send_Message with arguments '09' for Msg# and 'E' for MsgCode
OTHERWISE
Call Format_and_Send_Message with arguments '10' for Msg# and 'E' for MsgCode
END
```

## End Select VCBMORDER Routine;

```
Process_CBDVDUnitAddressesFile Routine:
```

```
Description: Process CB DVD Unit Addresses File.
Called by: Main Routine
Inputs: Action_Type_var
Outputs: none
   Parse Arg Action Type var
   SET Allocated Successfully = 'N'
   SET Max_Number_Try_alloc = 1
   DO WHILE (Allocated Successfully = 'N' & Max Number Try alloc <= cfg.MAX TRIES TO ALLOC)
      SET all rc = Allocate OLD File with arguments 'CBDVDUA' for DDname and
      'CB DVDUnitAddressesFileName var' for DSName.
      IF all rc = 0 THEN
      DO
          SET Allocated Successfully = 'Y'
          Call Format and Send Message with arguments '16' for Msg# and 'I' for MsgCode
          SET XX = 0
          SET MATCH DEVSTAT = 'N'
          SET CBDVDUA EOF = 'N'
          DO WHILE (MATCH_DEVSTAT = 'N' & CBDVDUA_EOF = 'N')
             SET XX = XX + 1
             "EXECIO 1 DISKRU CBDVDUA" Queue the next line on the stack
             IF rc = 2 THEN
                 SET CBDVDUA EOF = 'Y'
             PARSE PULL CBDVDUA REC
             SET DEVSTAT field = substr(CBDVDUA REC,6,8)
             IF Job Name var = DEVSTAT field THEN
                 SET MATCH DEVSTAT = 'Y'
          END
          IF MATCH DEVSTAT = 'N' THEN
          DO
             "EXECIO 0 DISKW CBDVDUA (FINIS" /*close the file*/
             "FREE FI(CBDVDUA)"
             Call Format_and_Send_Message with arguments '17' for Msg# and 'E' for MsgCode
          END
          ELSE
          DO
             SET CONNTYPE var = substr(CBDVDUA REC,5,1)
   IF (CONNTYPE var <> 'C' & CONNTYPE var <> 'I') THEN
   DO
   "EXECIO 0 DISKW CBDVDUA (FINIS" /*close the file*/
                    "FREE FI(CBDVDUA)"
                    Call Format and Send Message with arguments '30' for Msg# and 'E' for
                    MsgCode
                 END
             IF Action Type var = 'UPDATE STARTED' THEN
             DO
                 SET CB DVDUnitAddress var = substr(CBDVDUA REC,2,3)
```

```
SET parse value GetDateTime('DB2') with myrc Timestamp var
                 SET Time Started timestamp var = SUBSTR(Timestamp var,1,19)
                 IF myrc <> 0 THEN
                 DO
                     "FREE FI(CBDVDUA)"
                     Call Format and Send Message with arguments '18' for Msg# and 'E' for
                     MsgCode
                 END
                 SET CBDVDUA REC NEW= substr(CBDVDUA REC,1,34) ||
                 Time Started timestamp var
                 PUSH CBDVDUA REC NEW
                 Call Write_File with argument 'CBDVDUA' for DDname
             END
              ELSE
             IF Action Type var = 'UPDATE COMPLETED' THEN
              DO
                 SET CBDVDUA REC NEW= substr(CBDVDUA REC,1,5) || 'FREE'
                 PUSH CBDVDUA REC NEW
                 Call Write File with argument 'CBDVDUA' for DDname
             END
              Call Format_and_Send_Message with arguments '19' for Msg# and 'I' for MsgCode
          END
       END
       ELSE
       DO
          Call Format and Send Message with arguments '11' for Msg# and 'I' for MsgCode
          SET Max Number Try alloc = Max Number Try alloc + 1
          'SLEEP' cfg.SLEEP_SECONDS
       END
   END
   IF Max_Number_Try_alloc > cfg.MAX_TRIES_TO_ALLOC THEN
   DO
       IF Action Type var = 'UPDATE STARTED' THEN
          Call Format_and_Send_Message with arguments '20' for Msg# and 'E' for MsgCode
       IF Action_Type_var = 'UPDATE COMPLETED' THEN
          Call Format_and_Send_Message with arguments '20' for Msg# and 'W' for MsgCode
   END
End Process_CBDVDUnitAddressesFile Routine;
Process_CBIPConnectedDVDTowerDetailsFile Routine:
Description: Process CB IP Connected DVD Tower Details File.
Called by: Main Routine
Inputs: CB DVDUnitAddress parm
Outputs: DD Name for CB DVDTowerUnitNumber var
   Parse Arg CB_DVDUnitAddress_parm
SET MATCH CBDVDUN = 'N'
SET Ind = 1
   Call Read File with arguments 'IPCFG' for DDName, 'IPCFG' REC.' for StemName and 'N' for
   Free parm.
   IF IPCFG REC.0 > 0 THEN
   DO
```

```
DO WHILE (Ind <= IPCFG REC.0 & MATCH CBDVDUN ='N')
   SET CBDVDUN var = SUBSTR(IPCFG REC.ind,11,3)
   SET RECTYPE var = SUBSTR(IPCFG REC.ind,1,1)
   IF (CBDVDUN_var = CB_DVDUnitAddress_parm & RECTYPE_var ='U') THEN
   DO
SET MATCH CBDVDUN = 'Y'
   DD Name for CB DVDTowerUnitNumber var = SUBSTR(IPCFG REC.ind,2,8)
SET Ind = Ind + 1
   END
   END
   IF MATCH CBDVDUN = 'N' THEN
   Call Format and Send Message with arguments '29' for Msg# and 'E' for MsgCode
   END
End Process CBIPConnectedDVDTowerDetailsFile Routine;
Allocate OLD File Routine:
```

Description: Generic routine to allocate a OLD file using the derived file name.

Called by: Process CBDVDUnitAddressesFile routine

Inputs: DDName parm DSNAME parm

Outputs: rc

If error, CBDMCMD RC of 12, error mail notification and SYSPRINT log message

```
Parse Arg DDName_parm, DSNAME_parm
ADDRESS TSO "ALLOCATE DA(""DSNAME parm"") F("DDName parm") OLD"
SET DSNAME var = DSNAME parm
Return(rc)
```

#### End Allocate OLD File Routine;

# Process\_Type\_4 Routine:

```
Description: Process type 4 record.
Called by : Main Routine
Inputs: none-Type 4 Count, cfg.stem, CBPDO order, CBORDPR Tp4 REC.Type 4 This Count
Outputs: none-VOLSER field, Type 4 Processed, Total VOL Created, Process Vol.
Sysprint message and error email notifications
```

```
IF Type 4 Count = 0 THEN
      Call Format_and_Send_Message with arguments '08' for Msg# and 'E' for MsgCode
Set Process Vol = 'Y'
   DO Type 4 This Count = 1 TO Type 4 Count while(Process Vol = 'Y')
   Found DVD Media Type = 'N'
DO BB = 1 to WORDS(Cfg.DVD MEDIA TYPES)
IF CBPDO order = 'Y' THEN
             IF SUBSTR(CBORDPR Tp4 REC.Type 4 This Count,33,4) =
WORD(Cfg.DVD_MEDIA_TYPES,BB) THEN
```

```
Found_DVD_Media_Type = 'Y'
             ELSE
                    NOP
             END
      ELSE
             IF SUBSTR(CBORDPR Tp4 REC.Type 4 This Count,32,4) =
WORD(Cfg.DVD_MEDIA_TYPES,BB) THEN
             Found DVD Media Type = 'Y'
             END
      END
END
   IF Found DVD Media Type = 'Y' THEN
       IF CBPDO order = 'Y' THEN
          SET VOLSER_field = SUBSTR(CBORDPR_Tp4_REC.Type_4_This_Count,26,6)
       ELSE
          SET VOLSER field = SUBSTR(CBORDPR Tp4 REC.Type 4 This Count,25,6)
       SET CB DVDOrderPREMASTImageFileName var = cfg.CB DVD ORD FILE HLQ || '.' ||
       VCBMORDER IMFGRUN# || '.' || 'V' || VOLSER field || '.' ||,
                                                    cfg.CB_DVD_ORD_PREMAST_FILE_LLQ
       IF cfg.CALL CDWRITE = 'N' THEN
          Call Format and Send Message with arguments '23' for Msg# and 'I' for MsgCode
       Call Format and Send Message with arguments '24' for Msg# and 'I' for MsgCode
       Call Format and Send Message with arguments '24' for Msg# and 'I' for MsgCode
       SET CD rc = 0
       IF cfg.CALL CDWRITE = 'Y' THEN
          SET CDWRITEModule = """ || strip(Cfg.CDWRITE_PGM_NAME_LOAD) || """
          SET Parms = """ | strip(CB DVDOrderPREMASTImageFileName var) | ,
          " " || "DD:" || strip(efg.CB_DVD_UNIT_DD_NAME
          DD Name for CB DVDTowerUnitNumber var) | " " | "VOLUME=" | strip(VOLSER field)
          Address TSO "CALL " CDWRITEMODULE Parms
          SET CD rc = rc
       END
       IF CD_rc = 0+cfg.CALL_CDWRITE = 'N' THEN
       DO
          SET Total VOL Created = Total VOL Created + 1
          Call Format and Send Message with arguments '25' for Msg# and 'I' for MsgCode
       END
       ELSE
       DO
          Call Format_and_Send_Message with arguments '26' for Msg# and 'E' for MsgCode
          SET Process Vol = 'N'
       END
       END
       ELSE
       Call Format_and_Send_Message with arguments '27' for Msg# and 'I' for MsgCode
       END-IF
```

```
SET Type 4 Processed = Type 4 Processed + 1
END
ELSE
   SET Type 4 Bypassed = Type 4 Bypassed + 1
END
```

# End Process\_Type\_4 Routine;

# Write File Routine:

Description: Writes the records

Called by: Process CBDVDUnitAddressesFile Routine

Inputs: DDNameW StemName

Outputs: CBDMCMD\_RC.IF error, return\_code of 12, error email notification & SYSPRINT log message

Parse Arg DDNameW StemName

"EXECIO 1 DISKW "|| DDNameW || " (FINIS"

IF rc <> 0 THEN

Call Format and Send Message with arguments '15' for Msg# and 'E' for MsgCode

Drop (StemName) "FREE FI("DDNameW")"

# End Write File Routine;

#### Allocate CBDVDUnitAddress Routine:

Description: Allocate CBDVDUnitAddress

Called by: Main routine

Inputs: CBDVD DD p CBDVD UA p, CONNTYPE var

Outputs: rc

Parse Arg CBDVD\_DD\_p,CBDVD\_UA\_p

IF CONNTYPE var = 'C' THEN

ADDRESS TSO "ALLOCATE F("CBDVD DD p") NEW KEEP UNIT(" || CBDVD UA p || ")"

ADDRESS TSO "ALLOCATE F("CBDVD DD p") NEW KEEP UNIT(" || CBDVD UA p || ") SPACE(1,1) CYL"

## End Allocate CBDVDUnitAddress Routine;

# Format\_and\_Send\_Message Routine:

Description: Format and Send Message on SYSPRINT and email notification

Called by : Main, Read File, Read CTL File, Select VCBMORDER, Process Type 4,

Send Notification,

Allocate OLD File, Process CBDVDUnitAddressesFile, Write File, Program End,

Process CBIPConnectedDVDTowerDetailsFile routines

Inputs: Msg# MsgCode

Outputs: Sysprint message and error email notification

```
SET Send_mail = 'Y'
SET End process = 'Y'
SET Msg line = 1
SET Notify DD = 'ERRNTFY'
IF MsgCode = 'E' THEN
   SET CBDMCMD RC = 12
ELSE
IF MsgCode = 'W' THEN
DO
   SET CBDMCMD RC = 4
   SET Notify DD = 'WARNTFY'
END
IF (Msg# = '01' & MsgCode = 'E') THEN
   SET Msg_Text1 = 'NO RECORDS TO PROCESS IN CB MFG PRODUCTS FILE'
IF (Msg# = '02' & MsgCode = 'E') THEN
   SET Msg_Text1= 'REXX Error' rc 'in line' lin_var':' errortext(rc) || '. Source line=' sourceline(sigl)
ELSE
IF (Msg# = '03' & MsgCode = 'E') THEN
   SET Msg_Text1= 'REXX Error: Uninitialized variable' condition('D') "on line" lin_var
ELSE
IF (Msg# = '04' & MsgCode = 'E') THEN
   SET Msg Text1 = 'READ UNSUCCESSFUL FOR' DDName var 'DD'
ELSE
IF (Msg# = '05' & MsgCode = 'E') THEN
DO
   SET Msg Text1= 'ERROR TRYING TO CONNECT TO QMANAGER' Cfg.QMgr
   SET Send mail = 'N'
END
ELSE
IF (Msg# = '06' & MsgCode = 'E') THEN
DO
   SET Msg Text1= 'Error in CfgValidate - Invalid CFG input' | | ,
   ' parameters'
   SET Send mail = 'N'
END
ELSE
IF (Msg# = '07' & MsgCode = 'E') THEN
DO
   SET Msg_Text1= 'Error in CfgRead -' || CfgDD || ' File open failed'
   SET Send_mail = 'N'
END
ELSE
IF (Msg# = '08' & MsgCode = 'E') THEN
DO
   SET Msg Text1= 'NO TYPE 4 RECORDS FOUND FOR CB MFG ORDER #: ' ||
   CB Mfg Order field
   SET End process = 'N'
END
ELSE
IF (Msg# = '09' & MsgCode = 'E') THEN
   SET Msg Text1= 'NO ROW FOUND IN VCBMORDER FOR CB MFG ORDER#: ' ||
   VCBMORDER_IMFGORD
ELSE
```

```
IF (Msg# = '10' & MsgCode = 'E') THEN
   SET Msg_Text1= 'DB2 ERROR OCCURRED TRYING TO SELECT FROM VCBMORDER FOR
   CB MFG ORDER#: VCBMORDER IMFGORD
ELSE
IF (Msg# = '11' & MsgCode = 'I') THEN
DO
   SET Msg Text1= 'CB DVD Unit Addresses File ' || DSNAME var || ' could not be allocated for
   Action Type ' || Action_Type_var || ' will try and re-allocate in ' || cfg.SLEEP_SECONDS || '
   seconds'
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '12' & MsgCode = 'E') THEN
DO
   SET Msg Text1= 'CB$PREF failed with RC=' Process Notif rc
   SET Send mail = 'N'
END
ELSE
IF (Msg# = '13' & MsgCode = 'E') THEN
   SET Msg Text1= 'ERROR TRYING TO PUT MESSAGE TO OMSADMINQNAME ' ||
   Cfg.OMSADMINQName
   SET Send mail = 'N'
END
ELSE
IF (Msg# = '14' & MsgCode = 'E') THEN
DO
   SET Msg Text1= 'ERROR TRYING TO OPEN OMSADMINQNAME ' || Cfg.OMSADMINQName
   SET Send mail = 'N'
END
ELSE
IF (Msg# = '15' & MsgCode = 'E') THEN
   "FREE FI(CBDVDUA)"
   SET Msq Text1= 'Error occurred updating record in ' || CB DVDUnitAddressesFileName var ' for
   DVD Unit Address ' | CB DVDUnitAddressesFileName var | ' and Action Type ' |
   Action Type var
   SET Msg Text2= 'Error occurred updating record in ' || CB DVDUnitAddressesFileName var
END
ELSE
IF (Msg# = '16' & MsgCode = 'I') THEN
   SET Msg_Text1 = 'CB DVD Unit Addresses File ' || DSNAME_var || ' allocated for Action Type ' ||
   Action Type var
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '17' & MsgCode = 'E') THEN
   SET Msg Text1 = 'Unable to locate record for ' || Job Name var || ' in file ' ||
   CB DVDUnitAddressesFileName var
ELSE
IF (Msg# = '18' & MsgCode = 'E') THEN
   SET Msg Text1 = 'CBMDYT FAILED WITH RC=' MYRC
IF (Msg# = '19' & MsgCode = 'I') THEN
```

```
DO
   SET Msg Text1 = 'Record for DVD Unit Address' | CB DVDUnitAddress var | 'and Action
   Type ' || Action_Type_var || ' was updated successfully in CB DVD ' ||,
'Unit Addresses File ' | CB DVDUnitAddressesFileName var
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '20') THEN
   SET Msg Text1 = 'CB DVD Unit Addresses File ' || CB DVDUnitAddressesFileName var
   SET Msg Text2 = 'could not be allocated for Action Type ' || Action Type var || ' after ' ||
   cfg.MAX TRIES TO ALLOC || ' tries'
   SET Msg line = 2
   IF MsgCode = 'W' THEN
       SET End process = 'N'
END
ELSE
IF (Msg# = '21' & MsgCode = 'I') THEN
   SET Msg Text1 = 'DD Name ' || DD Name for CB DVDTowerUnitNumber var || ' and DVD Unit
   Address ' | CB DVDUnitAddress var | ' was allocated successfully'
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '22' & MsgCode = 'E') THEN
   SET Msq Text1 = 'DD Name ' || DD Name for CB DVDTowerUnitNumber var || ' and DVD Unit
   Address ' || CB DVDUnitAddress var || ' was not allocated successfully'
SET Final CB DVDUnitAddressesFileUpdate var = 'N'
   SET End process = 'N'
ELSE
IF (Msg# = '23' & MsgCode = 'I') THEN
   SET Msg Text1 = 'Bypassing call to CDWRITE'
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '24' & MsgCode = 'I') THEN
DO
   SET Msg Text1 = 'Calling CDWRITE for CB DVD Order ' || 'PREMAST Image File ' ||
   CB DVDOrderPREMASTImageFileName_var ||,
' using DD Name ' || DD_Name_for_CB_DVDTowerUnitNumber_var || ' and CB DVD Unit Address ' ||
CB DVDUnitAddress var
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
IF (Msg# = '25' & MsgCode = 'I') THEN
DO
   SET Msg Text1 = 'CDWRITE processed CB DVD Order PREMAST Image File ' ||
   CB DVDOrderPREMASTImageFileName var | | 'successfully
   SET Send mail = 'N'
   SET End process = 'N'
END
ELSE
```

```
IF (Msg# = '26' & MsgCode = 'E') THEN
   DO
       SET Msg Text1 = 'CDWRITE encountered an error and ended with an RC of ' | CD rc | '
       processing CB DVD Order PREMAST Image File ' ||,
                          CB DVDOrderPREMASTImageFileName var
       SET Msq Text2 = 'CDWRITE encountered an error and ended with an RC of ' || CD rc
   SET Final CB DVDUnitAddressesFileUpdate var = 'N'
       SET End process = 'N'
   END
   ELSE
   IF (Msg# = '27' & MsgCode = 'I') THEN
SET Msg Text1 = 'CDWRITE bypassed processing CB DVD Order PREMAST Image File ' ||
CB DVDOrderPREMASTImageFileName var
       SET Send mail = 'N'
       SET End process = 'N'
   END
   ELSE
   IF (Msg# = '28' & MsgCode = 'E') THEN
   SET Msg Text1 = 'Order#: ' || VCBMORDER IESWORDER || ', Line Item#: ' ||
   VCBMORDER_IESWLITEM || ', CB Mfg. Order#: ' || VCBMORDER_IMFGORD# || ', CB Mfg. Run#: '
   VCBMORDER IMFGRUN# || ' has been cancelled and no DVDs will be created. The DVDCRT
   Action for the order should be set to a status of Cancel.'
   SET Final CB DVDUnitAddressesFileUpdate var = 'N'
   END
   FLSE
   IF (Msg# = '29' & MsgCode = 'E') THEN
   SET Msg_Text1 = 'Unable to locate record for ' || CB_DVDUnitAddress_var ||,
   ' in the CB IP Connected DVD Tower Details File (IPCFG DD)'
   SET Final CB DVDUnitAddressesFileUpdate var = 'N'
   END
   ELSE
   IF (Msg# = '30' & MsgCode = 'E') THEN
   DO
   SET Msq Text1 = 'Connection Type field value in a CB DVD Unit Addresses File ' ||
   CB DVDUnitAddressesFileName var | | ' record not I or C'
   SET Final CB DVDUnitAddressesFileUpdate var = 'N'
   END
   ELSE
   IF (Msg# = '99' & MsgCode = 'I') THEN
   DO
       SET Msg Text1= 'CBDMCMD terminated with rc=' CBDMCMD RC
       SET Send mail = 'N'
       SET End process = 'N'
   END
   SET Code Desc = 'CBDMCMD0' || Msg# || MsgCode || ': '
   Call LogMsg Strip(value('Msg#'),,"""), value('MsgCode'), value('Msg_Text1')
   IF Msg# = '20' THEN
       Say Msg Text2
```

```
IF Send mail = 'Y' THEN
   DO
       SET Memo Body.1 = Code Desc | Msg Text1
       IF Msg# = '20' THEN
       DO
           SET Memo Body.2 = Msg Text2
           SET Dynamic Text = Msg Text1 || ' not allocated'
       END
       ELSE
       IF Msg# = '26' | Msg# = '15' THEN
           SET Dynamic Text = Msg Text2
       ELSE
           SET Dynamic Text = Msg Text1
       SET Memo Body.0 = Msg line
       Call Send Notification with 'Notify DD' argument for Notification DDname file
   END
   IF (Msg# = '28' | Msg# = '29' | Msg# = '30' ) Then
   CALL Program End
   IF End process = 'Y' THEN
       signal ExitPt
End Format and Send Message Routine;
Send Notification Routine:
Description: Calls Process email Notification File included routine which parses the email Notification
               if needed, converts special character for code page processing, and writes the email
               to a Process Notif eMail IDs array and provides the array
               Process Notif eMail IDs Count.
               It calls BuildMetaData which sets up MetaDataMsg structure for input into the OMSBM
               included routine, BuildMetaData, which formats the OMS Build Meta Data Message.
               The formatted metadata message is later sent to OMS in the $MQI included routine,
               MgiPutMsg..
Called by: Format and Send Message Routine
Input: cfg.EMAIL_ID_CHAR, Dynamic_Text, Memo_Body.
Outputs: If error, CBDMCMD RC of 12 SYSPRINT log message
   Parse Arg Notify DD p
   Call Process_Email_Notification_File(Notify_DD_p Cfg.EMAIL_ID CHAR)
   IF Process Notif rc = 0 THEN
   DO
       SET MD.iMsg. vars = "Application AdminMsgNo ProgramName MemoSubject DynamicText
       MemoTargets MemoBody Monospaced"
       SET MD.iMsg._Application = 'CB'
       IF Notify DD p = 'ERRNTFY' THEN
       DO
           SET MD.iMsg. MemoSubject = 'Create CB DVD Order Media Item Data (CBDMCMD)
           process error occurred'
```

SET MD.iMsg. AdminMsgNo = 4124

**END** 

```
ELSE
       DO
          SET MD.iMsg. MemoSubject = 'Create CB DVD Order Media Item Data (CBDMCMD)
          process warning occurred'
          SET MD.iMsg. AdminMsgNo = 4125
       END
       SET MD.iMsg. ProgramName = 'CBDMCMD'
       SET MD.iMsg. DynamicText = Dynamic Text
       DO M = 1 to Process Notif eMail Ids.0
          SET MD.iMsg. MemoTargets.M = Strip(Process Notif eMail Ids.M)
       END
       SET MD.iMsg. MemoTargets.0 = Process Notif eMail Ids.0
       DO ZZ = 1 to Memo Body.0
          SET MD.iMsg. MemoBody.ZZ = Memo Body.ZZ
       END
       SET MD.iMsg. MemoBody.0 = Memo_Body.0
       SET MD.iMsg. Monospaced = "Y"
       SET MessageText = BuildMetaData("iMsg")
       IF strip(MessageText) = " THEN
          SET CBDMCMD RC = 12
       ELSE
          Call Send_MQMessage_to_OMS
   END
   ELSE
       Call Format_and_Send_Message with arguments '12' for Msg# and 'E' for MsgCode
End Send Email Notification Routine;
Send MQMessage to OMS Routine:
Description: Invokes the following $MQI included routines to send the email message to OMS:
               - MqiOpenQueue opens the OMS input queue. It returns a value in Queuehandle
               valueIF it completes
               successfully and IF it does not complete successfully, it logs the error and returns a null
               value in
               QueueHandle.
               - MqiPutMsq puts the email message to the OMS MQ queue specified in the the
               OMSAdminQName
               parameter value in the configuration file.
               - MqiCloseQueue - which closes the input queue.IF error occurs, it logs the error.
Called by: Send Notification
Input: Cfg.OMS CODEPAGE, cfg.OMSAdminQName, MessageText
Outputs: IF system error, CBDMCMD_RC of 12, error email notification & SYSPRINT log message
   QueueHandle = MgiOpenQueue(cfg.OMSAdminQName, MQOO OUTPUT)
   IF QueueHandle <> " THEN
   /* Sets up message structure for the MqiPutMsg included routine */
   DO
       SET ipmd.MSG = MQMT DATAGRAM
       SET ipmd.PER = MQPER PERSISTENT
       SET ipmd.CCSI = Cfg.OMS CODEPAGE
       SET Myrc = MaiPutMsa (QueueHandle, Cfg.OMSADMINQName, MessageText, "ipmd.", "opmd.",
       MQPMO NO SYNCPOINT)
       IF Myrc = 0 THEN
          SET Myrc = MqiCloseQueue(QueueHandle, Cfg.OMSADMINQName)
```

Call Format\_and\_Send\_Message with arguments '13' for Msg# and 'E' for MsgCode

ELSE

END ELSE

Call Format\_and\_Send\_Message with arguments '14' for Msg# and 'E' for MsgCode

# End Send\_MQMessage\_to\_OMS Routine;

```
Program End Routine:
Description: Write summary report to SYSPRINT.
Called by: Main routine, Format and Send Message
Input: Type 1 Count, Type 4 Count, Total Discard Msg, Total CBORDPR Count,
Total ITMCONT Created.
              Total MEDITEMREQ Created, Total MEDITEMREP Created, Total VOL Created
              Type 4 Processed, Type 4 Bypassed
Outputs: Sysprint message
   IF Final CB DVDUnitAddressesFileUpdate var = 'Y' THEN
   SET Action Type var = 'UPDATE COMPLETED'
   Call Process CBDVDUnitAddressesFile with argument 'Action Type var' for Action Type variable.
   IF cfg.CALL CDWRITE = 'Y' THEN
   DQ
   IF CB DVDUnitAddressDDNameAllocated var = 'Y' THEN
   "FREE FI("cfg.CB_DVD_UNIT_DD_NAMEDD_Name_for_CB_DVDTowerUnitNumber_var")"
   END
   SET Total CBORDPR Count = Type 1 Count
   SAY (' ')
   SAY (' ')
   SAY ('CREATE CB DVD ORDER MEDIA SUMMARY REPORT')
   SAY (' ')
   SAY LEFT('TOTAL CB MFG PRODUCTS FILES PROCESSED: '.60) Total CBORDPR Count
   SAY LEFT('TOTAL TYPE 4 RECORDS PROCESSED:', 60) Type 4 CountType 4 Processed
   SAY LEFT('TOTAL TYPE 4 RECORDS BYPASSED: '.60) Type 4 Bypassed
   SAY LEFT(' TOTAL DVD VOLUMES CREATED:',60) Total VOL Created
   SAY (' ')
End Program_End Routine;
CBDMCMD EXECUTING JCL used for PIDB:

    Suggested executing JCL that can be used in SDTSOR.UTFT.EXECJCL

//S1 EXEC CBDMCMD.
// PROCLIB='SCLMPROD.STAGING.PROCLIB',
// EXECLIB='SCLMPROD.STAGING.CEXEC',
// CBASHLQ='CBSOMAT.UT',CBSOMAP=CBSOMAT
// SUMMARY=nnnnnn where nnnnnn = the test Products File created
//*
```

#### CBDMCMD PROC used for PIDB:

- The following is the JCL (to be located in **SCLMPROD.PROD.PROCLIB** and be moved to **CBSOMAP.PROCLIB**) that is used to run the CBDMCMD program on PIDZ and MVSZ: //CBDMCMD PROC CBASHLQ='BLDSERV.SOMAP',

```
// PROCLIB='CBSOMAP.PROCLIB',
// EXECLIB='BLDSERV.SOMAP.CEXEC',
// CBSOMAP=CBSOMAT,
// OUT=*
//S1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSEXEC DD DSN=&EXECLIB,DISP=SHR
//CTL DD DSN=&CBASHLQ..CTL.CFG(CBDMCMD),DISP=SHR
//CBORDPR DD DSN=&SUMMARY.,DISP=SHR <== SUMMARY set in Phase VA job that creates Action
Server Control File
//IPCFG DD DSN=&CBASHLQ..DVD.TOWER.DETAILS,DISP=SHR
//ERRNTFY DD DSN=&CBASHLQ..DVD.ERROR.EMAIL.NOTIFY,DISP=SHR
//WARNTFY DD DSN=&CBASHLQ..DVD.WARN.EMAIL.NOTIFY,DISP=SHR
//SYSTSPRT DD SYSOUT=&OUT
//SYSTSIN DD DSN=&PROCLIB(CBDMCMD1),DISP=SHR
- The following is the suggested info to set up in the CBDMCMD1 PROC:
%CBDMCMD
```

# **CBDMCMD SKEL used for PIDZ and MVSZ:**

- See the CBDMCMD SKEL used for PIDZ and MVSZ section in the CBDMCMD System Model for the SKEL details.