In general, the “Table Tool” – ENBPIU00 – can be used as a replacement for the EDVBind utility. Presented in this document is a working solution created for a provided example EDVBIND table. Features and considerations for this example solution include:

1. The EDVBIND table tested with this solution does not match the descriptions of the table provided in the EDVBIND documentation. Differences include:
	1. These columns are found in the EDVBIND documentation, but not found in example EDVBIND table: Environment, Suffix, ISO, RELEASE, EXPLAIN, VALIDATE, OTHER1, OTHER2, DDNAME
	2. These columns are found in the example EDVBIND table, but not mentioned in the EDVBIND documentation: LPar-SYSAFF, Back-Bind Stage and DBRM libraries
2. The EDVBIND utility submits jobs to do binds. The Table Tool solution can be adjusted to consistently submit jobs to do binds. However, this example will submit jobs only when the LPar-SYSAFF is not the one where the solution is running. This difference allows local binds to be directed to a separate output and executed within an Endevor processor. Then the DB2 Bind listing output can be captured within the element listing. An exception is made for Back Binds. It seems best that a production environment processor action would not need to wait for Back Binds to be completed in lower Environments. Outputs from the Table Tool replacement include:
	1. SUBMIT – content includes JCL with embedded Bind statements to be submitted for Binds on other Lpars. In a processor, SUBMIT is written to the internal reader. Jobcard statements (“/\*JOBPARM SYSAFF=&SYSAFF”) route Binds to other Lpars.
	2. BINDS – for DB2 Bind Commands that can be run on the current Lpar. Only Bind statements are written to BINDS, and no JCL statements are included. In a processor, route BINDS to a subsequent processor step (not included in the example Table Tool replacement) to execute the Bind statements.
3. EDVBIND runs separate steps for PLN binds and PKG binds. Each step also handles Back Binds. This Table Tool example replacement will build/submit PLN binds and PKG binds in a single step, but requires a second step for Back Binds. In this solution, package and Plan binds are built in the DB2BIND1 step. If any back binds are identified in the DB2STEP1 step, then its return code is a 1, and the DB2BIND2 step is run.
4. The example EDVBIND Table referenced DBRM libraries for Back Binds only. Up to four DBRM libraries are supported in this example Table Tool solution. In cases where fewer than four DBRM libraries are named on the table, the solution utilizes the “DUMMY\_DBRM” library.
5. EDVBIND documentation references PKG and PLN objects created by CONWRITE and used as EDVELM by a EDVBIND step. These objects are referenced as “MODELs” by the Table Tool and given DDnames as follows:
	1. MODLPKGR – the JCL for remote Package Binds
	2. MODLPKG - the Bind statement syntax for Package Binds
	3. MODLPLNR - the JCL for remote Plan Binds
	4. MODLPLN - the Bind statement syntax for Plan Binds
6. Some features of EDVBIND were not encountered in the example EDVBIND table, and are not supported by this Table Tool example, including:
	1. The “PKL” option
	2. The Log dataset
	3. The use of the “?” character as a mask

Processor steps from the Example Table Tool Solution:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// EXPORT SYMLIST=(\*)

//\*----- / test Example

// SET C1SY=CUSTOMER

*Set values for parameters for a single execution*

// SET C1SU=MPRS

// SET C1ELE=VSMIGSRE

// SET C1TY=COB

// SET C1STAGE=STG

//\*-----

*Set to your site CSIQCLS0 dataset and*

*Enter the name of your EDVBIND table*

// SET MYLPAR='MVS1'

// SET CSIQCLS0=CAPRD.NDVR.EMER.CATSNDVR.CSIQCLS0

// SET TABLE=WALJO11.DB2BIND.TABLES(EDVBIND2)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//DB2BIND1 EXEC PGM=IRXJCL,

*Table Tool (ENBPIU00) Runs in both steps.*

*First is the DB2BIND1 step*

// PARM='ENBPIU00 PARMLIST'

//TABLE DD DISP=SHR,DSN=&TABLE

//PLNBINDS DD \*,SYMBOLS=JCLONLY

 C1SY= '&C1SY'

 C1SU= '&C1SU'

*These are assigned to values from the JCL or from Endevor.*

 C1ELE= '&C1ELE'

 C1TY= '&C1TY'

 C1STAGE= '&C1STAGE'

 myLPAR = '&MYLPAR'

\* fixed .....

 BindType= 'PLN'

//PARMLIST DD \* <- Table search driver - search PLN and PKG binds

 NOTHING NOTHING DEFAULTS 0

 MODEL BINDS PLNBINDS 0

 MODEL BINDS OPTIONS1 A

 MODEL BINDS PKGBINDS 0

 MODEL BINDS OPTIONS1 A

 NOTHING BACKBIND BBINDRVR 1

//\*\*\* / Field and positions of EDVBIND table data

*This data contains field names and starting and ending positions of each on the EDVPARM table. If your table contains different fields, then you must update the contents of this data. For our Example EDVBIND table:*

**tSYS 9 18**

**tSUB 21 28**

**tELE 31 38**

**tTYP 42 49**

**tSTAGE 50 57**

**tcomment 1 1**

**DB2ID 66 73**

**LPARSysAFF 75 82**

**SYSAFF 75 82**

**QUALFIER 85 92**

**OWNER 96 103**

**BackBindStage 104 111**

**BackBindDBRMs 112 155**

//POSITION DD DISP=SHR,DSN=WALJO11.DB2BIND.CTL(EDVBIND)

//DEFAULTS DD \* / Set Defaults and initialize values

 $delimiter = '|'

 $nomessages = 'Y'

 Userid = USERID();

 $Table\_Type = 'positions'

 matched = 0; /\* Start with assumption of nomatch \*/

 BBindDbrm. = ''

 BBindStage = ''

 ListBBndDBRMS = ''

 ListBBndStages= ''

 DB2Package = ''

 myLPAR = MVSVAR(SYSNAME)

 DUMMY\_DBRM = 'WALJO11.ENDEVOR.DBRMLIB'

*Set the name of a Dummy DBRM library here. When fewer than 4 libraries are needed in a concatenation, the Dummy library is used.*

\* Other fixed Bind statement values:

 VALIDATE = 'BIND'

 ISO = 'CS'

 RELEASE = 'COMMIT'

 EXPLAIN = 'YES'

*If these values are found on your EDVBIND table, then delete these lines, which are here to provide fixed values for all Binds*

//PKGBINDS DD \* / Set Table search for PKG binds

 $SkipRow = 'N'

 BindType= 'PKG'

//OPTIONS1 DD \* / Set Defaults and initialize values

$SkipRow = 'N'

If tcomment = '\*' | $row# < 1 then $SkipRow = 'Y'

\*

MODEL = 'MODL' || BindType

TBLOUT = 'BINDS'; +

If SYSAFF /= myLPAR then, +

 Do ; +

 MODEL= MODEL || 'R' ; +

 TBLOUT = 'SUBMIT' ; +

 End;

If tSYS /= '+' & tSYS /= '.' then, +

 DO; matched = 1;+

 If (C1SY/=tSYS & tSYS/='\*') then matched = 0; +

 If (C1SU/=tSUB & tSUB/='\*') then matched = 0; +

 If (C1TY/=tTYP & tTYP/='\*') then matched = 0; +

 If (C1STAGE/=tSTAGE & tSTAGE/='\*') then matched = 0; +

 If (C1ELE/=tELE & tELE/='\*') then matched = 0; +

 DB2Package = SYSAFF || '.' || QUALFIER ; +

 If matched then x = BuildFromModel(MODEL); +

 End;

If matched then, +

 Do ; +

 Say C1SY C1SU C1TY C1Stage C1ELE 'matched with'; +

 Say tSYS tSUB tTYP tSTAGE tELE ; +

 If tSYS = '+' then x = BuildFromModel(MODEL); +

 If BackBindStage > ' ' then, +

 Do; +

 BBindStage = BackBindStage ; +

 If tSYS /= '.' & Wordpos(BBindStage,ListBBndStages)=0 then, +

 ListBBndStages = ListBBndStages BBindStage ; +

 End; +

 If BackBindDBRMs > ' ' then, +

 Do ;+

 ListBBndDBRMS = BBindDbrm.BBindStage ; +

 If Wordpos(BackBindDBRMs,ListBBndDBRMS) = 0 then, +

 Do; +

 ListBBndDBRMS = Strip(ListBBndDBRMS) BackBindDBRMs; +

 BBindDbrm.BBindStage = ListBBndDBRMS ; +

 TBLOUT = 'DB2BIND1'; +

 x = BuildFromModel(BACKSTIN); +

 End; +

 End; +

 End

$SkipRow = 'Y'

//BBINDRVR DD \* / Produce BackBind requests if any are found

 If Words(ListBBndStages) > 0 then $my\_rc =1;

 TBLOUT = 'BACKBIND' ;

 Do bbind# = 1 to Words(ListBBndStages) ; +

 x = BuildFromModel(BILDBBND); +

 End;

 $SkipRow = 'Y'

//BILDBBND DD \* / Setup search criteria for Back Binds (later step)

 MODEL BINDS PLNBINDS 0

 MODEL BINDS OPTIONS1 A

 MODEL BINDS PKGBINDS 0

 MODEL BINDS OPTIONS1 A

//BACKSTIN DD \* / Setup search criteria for Back Binds (later step)

BackBindSearch= '&C1SY &C1SU &C1ELE &C1TY'

ListBBndStages= '&ListBBndStages'

BackBindDBRMs.&BBindStage='&ListBBndDBRMS'

//MODLPKGR DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGR) <Rmot pkg Bind

// DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGE) < Package Bind

*These are the Table Tool ‘Models’ – painted output images. You can tailor yours to match what is needed at your site. See example contents below.*

//MODLPKG DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGE) < Package Bind

//MODLPLNR DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLNR) <Rmot PLN Bind

// DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLAN) <- Plan Bind

//MODLPLN DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLAN) <- Plan Bind

//SYSTSPRT DD SYSOUT=\*

//SYSTSIN DD DUMMY

//NOTHING DD DUMMY

//SYSEXEC DD DISP=SHR,DSN=&CSIQCLS0

*These are the outputs for the Table Tool step.*

 *In your processor, route BINDS to a temporary dataset and used by a subsequent BIND step. Also route SUBMIT to the internal reader ie….*

*//SUBMIT DD SYSOUT=(A,INTRDR)*

//BINDS DD SYSOUT=\*

//SUBMIT DD SYSOUT=\* (A,INTRDR)

//\*

//DB2BIND1 DD DSN=&&DB2BIND1,DISP=(MOD,PASS),

// UNIT=SYSDA,SPACE=(TRK,(5,5)),

*Back Bind information is collected by the DB2BIND1 step and provided to the DB2BIND2 step here.*

// DCB=(RECFM=FB,LRECL=240,BLKSIZE=24000)

//\*ACKBIND DD SYSOUT=\*

//BACKBIND DD DSN=&&BACKBIND,DISP=(MOD,PASS),

// UNIT=SYSDA,SPACE=(TRK,(5,5)),

// DCB=(RECFM=FB,LRECL=080,BLKSIZE=24000)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Table Tool (ENBPIU00) Runs in both steps.*

*This is the DB2BIN2 step for “Back” Binds*

//DB2BIND2 EXEC PGM=IRXJCL,COND=(1,NE,DB2BIND1),

// PARM='ENBPIU00 PARMLIST'

//TABLE DD DISP=SHR,DSN=&TABLE

//PARMLIST DD \* <- Driver

 NOTHING NOTHING DEFAULTS 0

 NOTHING NOTHING DB2BIND1 0

*Back Bind information is collected by the DB2BIND1 step and provided to the DB2BIND2 step here..*

// DD DSN=&&BACKBIND,DISP=(OLD,DELETE)

//DEFAULTS DD \* / Set Defaults and initialize values

 $delimiter = '|'

 $nomessages = 'Y'

 Userid = USERID();

*Set the name of a Dummy DBRM library here. When fewer than 4 libraries are needed in a concatenation, the Dummy library is used.*

 $Table\_Type = 'positions'

 matched = 0; /\* Start with assumption of nomatch \*/

 ListBBndDBRMS = ''

 DB2Package = ''

 myLPAR = MVSVAR(SYSNAME)

*Back Bind information is collected by the DB2BIND1 step and provided to the DB2BIND2 step here.*

 DUMMY\_DBRM = 'WALJO11.ENDEVOR.DBRMLIB'

//\* / Data passed from step DB2BIND1

//DB2BIND1 DD DSN=&&DB2BIND1,DISP=(OLD,DELETE)

//\*\*\* / Field and positions of EDVBIND table data

//POSITION DD DISP=SHR,DSN=WALJO11.DB2BIND.CTL(EDVBIND)

*This data contains field names and starting and ending positions of each on the EDVPARM table. If your table contains different fields, then you must update the contents of this data. It should match the POSITION data in the first step.*

//PLNBINDS DD \* / Set Table search for PLN binds & element data

 $SkipRow = 'N'

 BindType= 'PLN'

 C1SY = Word(BackBindSearch,1) ;

 C1SU = Word(BackBindSearch,2) ;

 C1ELE = Word(BackBindSearch,3) ;

 C1TY = Word(BackBindSearch,4) ;

 C1STAGE = Word(ListBBndStages,1)

 ListBBndStages = DELWORD(ListBBndStages,1,1)

 myDBRMLibs = BackBindDBRMs.C1STAGE DUMMY\_DBRM ;

 myDBRMLibs = myDBRMLibs DUMMY\_DBRM DUMMY\_DBRM DUMMY\_DBRM ;

 DBRM1 = Word(myDBRMLibs,1)

 DBRM2 = Word(myDBRMLibs,2)

 DBRM3 = Word(myDBRMLibs,3)

 DBRM4 = Word(myDBRMLibs,4)

*If these values are found on your EDVBIND table, then delete these lines, which are here to provide fixed values for all Binds*

\* Other fixed Bind statement values:

 VALIDATE = 'BIND'

 ISO = 'CS'

 RELEASE = 'COMMIT'

 EXPLAIN = 'YES'

//PKGBINDS DD \* / Set Table search for PKG binds

 $SkipRow = 'N'

 BindType= 'PKG'

//OPTIONS1 DD \* / Set Defaults and initialize values

 If tcomment = '\*' | $row# < 1 then $SkipRow = 'Y'

\*\* Always submit Back Binds...

 MODEL = 'MODL' || BindType || 'R'

 TBLOUT = 'SUBMIT' ;

\*\*

 If tSYS /= '+' & tSYS /= '.' then, +

 DO; matched = 1;+

 If (C1SY/=tSYS & tSYS/='\*') then matched = 0; +

 If (C1SU/=tSUB & tSUB/='\*') then matched = 0; +

 If (C1TY/=tTYP & tTYP/='\*') then matched = 0; +

 If (C1STAGE/=tSTAGE & tSTAGE/='\*') then matched = 0; +

 If (C1ELE/=tELE & tELE/='\*') then matched = 0; +

 DB2Package = SYSAFF || '.' || QUALFIER ; +

 If matched then, +

 Do ; +

 x = BuildFromModel(MODEL); +

 x = BuildFromModel(DBRMLIB); +

 End; +

 End;

 If matched & tSYS = '+' then, +

 Do ; +

 Say C1SY C1SU C1TY C1Stage C1ELE 'matched with'; +

 Say tSYS tSUB tTYP tSTAGE tELE ; +

 x = BuildFromModel(MODEL); +

 x = BuildFromModel(DBRMLIB); +

 End;

 $SkipRow = 'Y'

//MODLPKGR DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGR) <Rmot pkg Bind

// DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGE) < Package Bind

*These are the Table Tool ‘Models’ – painted output images. You can tailor yours to match what is needed at your site. See example contents below.*

//MODLPKG DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPKGE) < Package Bind

//MODLPLNR DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLNR) <Rmot PLN Bind

// DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLAN) <- Plan Bind

//MODLPLN DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(BINDPLAN) <- Plan Bind

*Back Bind information is collected from the EDVBIND table with named DBRM’s. This model places the named DBRMs into the submitted JCL. (Back Binds are always submitted).*

//DBRMLIB DD DISP=SHR,DSN=WALJO11.DB2BIND.JCL(DBRMLIB) <- DBRMLIBs

//SYSTSPRT DD SYSOUT=\*

//SYSTSIN DD DUMMY

//NOTHING DD DUMMY

//SYSEXEC DD DISP=SHR,DSN=&CSIQCLS0

*There is only one output for the DB2BIND2 step, since the step only addresses Back Binds, which are always submitted in batch. From the DB2BIND2 step there is no BINDS output.*

 *In your processor, route SUBMIT to the internal reader ie….*

*//SUBMIT DD SYSOUT=(A,INTRDR)*

//BINDS DD DUMMY

//SUBMIT DD SYSOUT=\* (A,INTRDR)

//BACKBIND DD SYSOUT=\*

Inputs for the example Table Tool solution:

Heading from the example EDVBIND table

----+----1----+----2----+----3----+----4----+----5----+----6----+----7 ----+----8----+----9----+----0----+----1----+----2----+---

\* SYSTEM SUBSYS ELEMENT TYPE STAGE DB2 LPAR QUALFIER OWNER STAGE DBRMS (FOR BACKBINDS ONLY)

\* SUB SYSAFF

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

If your EDVBIND table contains columns different from this one, then adjustments are required.

Table Tool Models:

Content of WALJO11.DB2BIND.JCL(BINDPKGR)

//&Userid|P JOB (6491099999999),'BIND PKG &DB2ID',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=&SYSAFF

//JOBLIB DD DSN=SYS1.&DB2ID.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPKGR) - &MODEL

//\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

Content of WALJO11.DB2BIND.JCL(BINDPKGE)

DSN SYSTEM(&DB2ID)

BIND PACKAGE(&DB2Package) -

 OWNER(&OWNER) -

 QUALIFIER(&QUALFIER) -

 MEMBER(&C1ELE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

Content of WALJO11.DB2BIND.JCL(BINDPLNR)

//&Userid|B JOB (6491099999999),'BIND PLN &DB2ID',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=&SYSAFF

//JOBLIB DD DSN=SYS1.&DB2ID.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPLNR) - - &MODEL

'/\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

Content of WALJO11.DB2BIND.JCL(BINDPLAN)

DSN SYSTEM(&DB2ID)

BIND PLAN(&C1ELE) -

 OWNER(&OWNER) -

 QUAL(&QUALFIER) -

 VALIDATE(&VALIDATE) -

 ISO(&ISO) -

 RELEASE(&RELEASE) -

 EXPLAIN(&EXPLAIN)

**Sample outputs:**

BINDS output from step DB2BIND1:

DSN SYSTEM(DBT1)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA4) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

DSN SYSTEM(DBQ1)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA4) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

DSN SYSTEM(DBT1)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA4) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

DSN SYSTEM(DBQ1)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA4) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

Submit output from step DB2BIND1:

//WALJO11B JOB (6491099999999),'BIND PLN DBP5',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=MVSD

//JOBLIB DD DSN=SYS1.DBP5.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPLNR) - - MODLPLNR

'/\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBP5)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA1) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

//WALJO11B JOB (6491099999999),'BIND PLN DBP5',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=MVSD

//JOBLIB DD DSN=SYS1.DBP5.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPLNR) - - MODLPLNR

'/\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBP5)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA2) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

//WALJO11P JOB (6491099999999),'BIND PKG DBP5',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=MVSD

//JOBLIB DD DSN=SYS1.DBP5.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPKGR) - MODLPKGR

//\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBP5)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA1) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

//WALJO11P JOB (6491099999999),'BIND PKG DBP5',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=MVSD

//JOBLIB DD DSN=SYS1.DBP5.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPKGR) - MODLPKGR

//\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBP5)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA2) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

Submit output from step DB2BIND2:

//WALJO11B JOB (6491099999999),'BIND PLN DBT1',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=CA06

//JOBLIB DD DSN=SYS1.DBT1.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPLNR) - - MODLPLNR

'/\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBT1)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA4) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

//DBRMLIB DD DISP=SHR,DSN=VLM.C1.MNTDBRM

// DD DISP=SHR,DSN=VLM.C1.MNQADBRM

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//WALJO11P JOB (6491099999999),'BIND PKG DBT1',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=CA06

//JOBLIB DD DSN=SYS1.DBT1.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPKGR) - MODLPKGR

//\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBT1)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA4) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

//DBRMLIB DD DISP=SHR,DSN=VLM.C1.MNTDBRM

// DD DISP=SHR,DSN=VLM.C1.MNQADBRM

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//WALJO11B JOB (6491099999999),'BIND PLN DBT1',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=CA06

//JOBLIB DD DSN=SYS1.DBT1.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPLNR) - - MODLPLNR

'/\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBT1)

BIND PLAN(VSMIGSRE) -

 OWNER(C1PLM) -

 QUAL(DBA4) -

 VALIDATE(BIND) -

 ISO(CS) -

 RELEASE(COMMIT) -

 EXPLAIN(YES)

//DBRMLIB DD DISP=SHR,DSN=VLM.C1.MNQADBRM

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//WALJO11P JOB (6491099999999),'BIND PKG DBT1',

// MSGCLASS=X,CLASS=A

/\*JOBPARM SYSAFF=CA06

//JOBLIB DD DSN=SYS1.DBT1.DSNLOAD,DISP=SHR

//\*--------------------------------------------------

//\*--- WALJO11.DB2BIND.JCL(BINDPKGR) - MODLPKGR

//\*--------------------------------------------------

//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20

//SYSTSPRT DD SYSOUT=\*

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//REPORT DD SYSOUT=\*

//SYSTSIN DD \*

DSN SYSTEM(DBT1)

BIND PACKAGE(CA06.DBA4) -

 OWNER(C1PLM) -

 QUALIFIER(DBA4) -

 MEMBER(VSMIGSRE) -

 ACTION(REPLACE) -

 RELEASE(COMMIT) -

 VALIDATE(BIND) -

 ISOLATION(CS) -

 EXPLAIN(YES) -

 CURRENTDATA(NO) -

 KEEPDYNAMIC(NO) -

 SQLERROR(NOPACKAGE) -

 NOREOPT(VARS) -

 DEGREE(1) -

//DBRMLIB DD DISP=SHR,DSN=VLM.C1.MNQADBRM

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

// DD DISP=SHR,DSN=WALJO11.ENDEVOR.DBRMLIB

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*