Advantage™ VISION:Report® Interface to DB2® for VSE

Installation Guide

16.1

Computer Associates™

ILINV161.PDF/D23-010-010

Contents

Chapter 1: MVS Installation and Use

Chapter 2: VSE Installation and Use

Index

Chapter

1 MVS Installation and Use

In this guide, the terms DB2® and SQL / DS™ are used interchangeably unless otherwise specified, although in general, the term DB2 is used in reference with MVS® and the term SQL / DS is used in reference with VSE.

Care has been taken to ensure that the MVS and VSE versions are compatible. However, there are certain SQL commands that are unique to the Operating System.

DB2 Authorization

Before beginning the installation of VISION:Report Interface to DB2, the database administrator for your site must authorize your use of DB2. VISION:Report Interface to DB2 utilizes an installed plan (refer to IBM's DB2 Reference Guide for a detailed description of a DB2 plan) that is linked to the supplied interface modules. This plan must be added to the DB2 plan library using the BIND command. It is then used by the interface to access DB2. In the following example, the DB2 plan name is QJ#PLAN.

To install the plan, the user submitting the VISION:Report Interface to DB2 installation job (as identified on the USER parameter of the JOB card) must have BIND authority on the DB2 plan name QJ#PLAN. This can be achieved by issuing the following SQL command under SPUFI:

```
GRANT BIND ON PLAN QJ#PLAN TO username;
```

In addition, EXECUTE authority must be granted to anyone wanting to use VISION:Report Interface to DB2. Specific EXECUTE grants can be made with the following SQL command from SPUFI:

```
GRANT EXECUTE ON PLAN QJ#PLAN TO username;
```

If everyone at your installation needs this authority, issue the following SQL command under SPUFI:

```
GRANT EXECUTE ON PLAN QJ#PLAN TO PUBLIC;
```

EXECUTE authority on QJ#PLAN allows you to run VISION:Report Interface to DB2. It does not permit access to any DB2 object (tables, columns, views, databases, etc.) that has not been explicitly granted to a particular user.

Number of Cursors

VISION:Report Interface to DB2 is linked with a CSECT table. This table contains all the information needed from the DB2 preprocessing of the output for the PRE#SQL macro, as described further in the QJOPTION and PRE#SQL Comparisons section in this chapter. For each possible cursor, approximately 20 bytes are added to the CSECT. You can generate the CSECT with entries from 1 to 999 cursors in the CNTLPREP member, as described in Installing VISION:Report Interface to DB2.

Customizing with the VISION:Report QJOPTION Macro

When VISION:Report Interface to DB2 is run with VISION:Report, the following keywords should be noted in the assembly of the QJOPTION macro:

SQLA1-SQLA5	Refer to VISION:Report Reference Guide, OPTION verb.
SQLPLNM	DB2 plan name. This keyword could be omitted, in which case the PLAN keyword in the PRE#SQL macro or OPTION statement can be used to override the value. This field is 1 to 8 characters long and has a default of ********.
SQLSYSN	DB2 system or subsystem name. This keyword could be omitted, in which case the SYSTEM keyword in the PRE#SQL macro or OPTION statement can be used to override the value. This field is 1-4 characters long and has a default of ****. See QJOPTION and PRE#SQL Comparisons.

Installing VISION:Report Interface to DB2

Installing VISION:Report Interface to DB2 requires running two jobs; the second job is run after the successful completion of the first job. The two jobs are:

- CNTLPREP
- CNTLINST

Note: Prior to running any of these jobs, you will need to change the JOB card to match your installation's requirements. In addition, the data set names for the VISION:Report libraries must match the data set names as originally installed. The data set name for your DB2 DBRMLIB must also be specified.

The following discussion describes these jobs.

The member, CNTLPREP, which was downloaded into the VISION:Report Interface to DB2 sample library (DB2SAMP), contains the steps necessary to create a CSECT, SQLCSECT, for cursors 1-999. CNTLPREP consists of the following steps:

ASMPRE	Assemble using PRE#SQL macro to specify DB2 system and plan names, as well as release and version number.
DB2PREP	Preprocess source code and create DBRM. Note that the DB2PREP step receives a condition code of 4 and issues two warning messages. You can ignore the messages.
DB2POST	Post-process DB2 preprocessed statements.
ASMPOST	Assemble DB2 post/preprocessed statements.
COPY	Copy output to VISION:Report object library.

The CNTLINST job links DB2 for VISION:Report Interface to DB2. Ensure that the DB2 load library data set name matches that of your installation. CNTLINST consists of the following steps:

LINKDB2	Links VISION:Report modules into DB2 load library.
BINDPLAN	Binds DB2 plan.

Make sure that the proper DB2 authorization has been established. (See DB2 Authorization.)

When you initially install VISION:Report, one of the job steps optionally downloads the VISION:Report Interface to DB2 sample library. If you have not downloaded this control library, do so now. It contains installation samples, such as CNTLPREP and CNTLINST, as well as test members.

Ensure that the sample JCL meets your installation requirements. In particular, the various data set names (DSN=) should match the data set names given to your DB2 libraries, and VISION:Report object and load libraries, as well as the plan member name (QJ#PLAN).

There are also several test members in the DB2 sample library. Review the contents of the member @INDEX, which contains a brief description of each test. These tests verify that the installation of VISION:Report Interface to DB2 was successful.

You need to run CNTLPREP first, followed by CNTLINST. Optionally, you can run the test job streams. The following are descriptions of the two members needed to complete your installation:

CNTLPREP — Sample JCL to create and code DBRMs. Creates a preprocessor input stream to VISION:Report Interface to DB2 post-processor. See CNTLPREP for a listing of CNTLPREP. In the first step, there is a job stream for assembling the various options using the PRE#SQL macro. Note that the DB2PREP step receives a condition code of 4 and issues two warning messages. You can ignore the messages. Options in the PRE#SQL macro are:

- SYSTEM — DB2 system name. There is no default. This is a 1-4 character field.

  Required unless you are going to use TSO Attach, or if you are always going to use the OPTION statement to override the value.

- PLAN — DB2 plan name. There is no default. This is a 1-8 character field.

  If the PLAN name is going to be the same as the DBRM name or if you are going to use TSO Attach, this field is not required.

- CURSORS — Number of cursors permitted. The default is 18; the allowable range is 1-999.

- RELEASE — DB2 release number. The default is 2.

- VERSION — DB2 version number. The default is 3.

- TYPE — Always use DB2.

CNTLINST — Sample JCL to link edit modules and bind DB2 plan. This step must run after CNTLPREP. See CNTLINST for a listing of CNTLINST.

CNTLPREP

```
//CNTLPREP JOB (ACCTINFO),MSGCLASS=X,REGION=2M,COND=(4,LT)
//*
//*********************************************************************
//* 1) GENERATE DYNAMIC SQL PROTOTYPE STATEMENTS
//* 2) PROCESS SQL STATEMENTS TO CREATE SQL ASSEMBLER STATEMENTS
//*    AND DBRM
//* 3) EXTRACT SQL PRAMATERS FROM SQL ASSEMBLER STAMENTS TO FORM
//*    ASSEMBLER SOURCE FOR SQL TABLE
//* 4) ASSEMBLE SQL TABLE
//* 5) COPY REBLOCKED SQL TABLE OBJECT TO OBJECT LIBRARY
//*********************************************************************
//*
//ASMPRE   EXEC PGM=ASMA90,REGION=3M,PARM=(TERM,OBJ)
//SYSLIN   DD   DISP=(,PASS),UNIT=DISK,SPACE=(80,(1500,250))
//SYSUT1   DD   UNIT=DISK,SPACE=(1700,(1000,250))
//SYSLIB   DD   DISP=SHR,DSN=@@@AAA.@@@BBB.MACLIB
//SYSPRINT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSIN    DD   *
        PRE#SQL SYSTEM=D41A,                                           -
               PLAN=QUIKJOB,                                           -
               CURSORS=20,                                             -
               VERSION=4,                                              -
               RELEASE=1
*  MAY USE     PLAN=,  TO DEFAULT TO DBRM MEMBER NAME
        END
/*
//*
//DB2PREP  EXEC PGM=DSNHPC,PARM='HOST(ASM),STDSQL(NO),SOURCE'
//STEPLIB  DD   DISP=SHR,DSN=SYS2.DB2410.SDSNLOAD
//SYSIN    DD   DISP=(OLD,DELETE),DSN=*.ASMPRE.SYSLIN
//SYSPRINT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSCIN   DD   DISP=(,PASS),UNIT=DISK,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=80,RECFM=FB,BLKSIZE=8160)
//SYSUT1   DD   UNIT=DISK,SPACE=(8160,(500,500))
//SYSUT2   DD   UNIT=DISK,SPACE=(8160,(500,500))
//DBRMLIB  DD   DISP=OLD,DSN=@@@AAA.@@@BBB.DBRMLIB(QUIKJOB)
//*
//DB2POST  EXEC PGM=POST#SQL
//STEPLIB  DD   DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB
//SYSPUNCH DD   DISP=(,PASS),UNIT=DISK,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=80,RECFM=FB,BLKSIZE=8160)
//SYSIN    DD   DISP=(OLD,DELETE),DSN=*.DB2PREP.SYSCIN
//SYSUDUMP DD   SYSOUT=*
//*
//ASMPOST  EXEC PGM=ASMA90,REGION=3M,PARM=(TERM,OBJ)
//SYSLIN   DD   DISP=(,PASS),UNIT=DISK,SPACE=(80,(1500,250))
//SYSUT1   DD   UNIT=DISK,SPACE=(1700,(1000,250))
//SYSLIB   DD   DISP=SHR,DSN=@@@AAA.@@@BBB.MACLIB
//SYSPRINT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSIN    DD   DISP=(OLD,DELETE),DSN=*.DB2POST.SYSPUNCH
//*
//COPY     EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=(OLD,DELETE),DSN=*.ASMPOST.SYSLIN
//SYSUT2   DD   DISP=OLD,DSN=@@@AAA.@@@BBB.OBJLIB(SQLCSECT)
//SYSIN    DD   DUMMY
//
```

CNTLINST

```
//CNTLINST JOB (ACCTINFO),MSGCLASS=X,REGION=2M,COND=(0,NE)
/*JOBPARM S=CA31
//*
//*********************************************************************
//* 1) LINK PROGRAM OBJECTS WITH SQL TABLE OBJECT TO CREATE
//*    EXECUTABLE MODULE
//* 2) BIND SQL PLAN
//* NOTE:
//* ADD 'INCLUDE DB2OBJ(DSNALI)' TO LINK TO FORCE TSO/IMS ATTACH.
//* SELECT DDF TO INCLUDE DDF FACILITIES IN BIND
//*********************************************************************
//*
//LINKDB2  EXEC PGM=IEWL,PARM='SIZE=(256K,64K),LET,LIST,XREF,MAP'
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=DISK,SPACE=(1024,(200,10))
//SYSLIB   DD   DISP=SHR,DSN=SYS2.DB2410.SDSNLOAD
//OBJ      DD   DISP=SHR,DSN=@@@AAA.@@@BBB.OBJLIB
//SYSLMOD  DD   DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB
//SYSLIN   DD   *
  INCLUDE       OBJ(QJESQL,QJO#SQL,DYL#SQL,DYL#CAF,SQLCSECT)
  NAME QUIKESQL(R)
/*
//*
//BINDPLAN EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DISP=SHR,DSN=SYS2.DB2410.SDSNLOAD
//DBRMLIB  DD   DISP=SHR,DSN=@@@AAA.@@@BBB.DBRMLIB
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DDNAME=NODDF
//NODDF    DD   *
  DSN SYSTEM(D41A)
  BIND PLAN(QUIKJOB) RETAIN MEMBER(QUIKJOB) ACTION(REPLACE)
  END
/*
//DDF      DD   *
  DSN SYSTEM(D41A)   /* PRIMARY LOCATION */
  BIND PACKAGE (LOCATION_001.RPT_L001) /* ONE PER LOCATION */ +
       MEMBER(QUIKJOB) ACTION(REPLACE)
  BIND PACKAGE (LOCATION_002.RPT_L002) /* ONE PER LOCATION */ +
       MEMBER(QUIKJOB) ACTION(REPLACE)
  BIND PACKAGE (LOCATION_003.RPT_L003) /* ONE PER LOCATION */ +
       MEMBER(QUIKJOB) ACTION(REPLACE)
  BIND PLAN(QUIKJOB) PKLIST(DB2LOCA.DYLDB2.*, VSELOCA.IBMUSER.*)  -
       PKLIST(LOCATION_001.RPT_L001.*, LOCATION_002.RPT_L002.*, +
              LOCATION_003.RPT_L003) +
       ACTION (REPLACE) RETAIN MEMBER (QUIKJOB)
  END
/*
//
```

QJOPTION and PRE#SQL Comparisons

The QJOPTION and PRE#SQL macros appear to have duplicate options. To avoid potential conflicts, the following describes the priorities between the macros.

QJOPTION	PRE#SQL	Action
No system or plan name (defaults to all *)	System and plan name	Takes system and plan Name from PRE#SQL
System and / or plan name	System and / or plan name	Takes system and / or plan name from QJOPTION
Within the VISION:Report program, if OPTION statement specifies: SQLSYSN=xxxx and / or SYSPLNM=xxxxxxxx	System and / or plan name	Takes system and / or plan name from OPTION statement first, then QJOPTION, if specified, then PRE#SQL macro

This method gives the greatest flexibility without requiring reassembly of the QJOPTION macro. It allows overrides using the OPTION statement or lets the option statement default to the PRE#SQL system and / or plan names.

QJTSTDB2

```
//QJTSTDB2 PROC REG=2M           REGION SIZE DEFAULT
//* **************************************************************
//*                                                              *
//*QJTSTDB2:                                                     *
//*                                                              *
//*          THIS PROC IS TO TEST VISION:INTERFACE FOR DB2       *
//*                                                              *
//* YOU CAN OVERRIDE THE REG(ION) SIZE                           *
//* YOU WILL NEED TO SUPPLY:                                     *
//*                                                              *
//*   SYSUT1, OFA, SYSIN, ETC.                                   *
//*                                                              *
//*  SOMETIMES, YOU MAY WANT TO OVERRIDE THE SLIB2               *
//*  WHICH IS THE DB2 LIBRARY                                    *
//*                                                              *
//*  OVERRIDE 'SLIB2'  FOR DB2 LIBRARY.  <<===================   *
//*                                                              *
//*  NOTE:  SOMETIMES YOU MAY NEED TO RUN TRACES.  IF THAT IS    *
//*         THE CASE, OVERRIDE THE "TR#SQL" DD STATEMENT         *
//*         TO BE "SYSOUT=*", RATHER THAN ITS CURRENT "DUMMY".   *
//*  CHANGE @@@AAA.@@@BBB TO REPORT LIBRARY                      *
//* **************************************************************
//*
//QJ       EXEC PGM=QUIKJOB,REGION=&REG
//STEPLIB  DD  DISP=SHR,DSN=@@@AAA.@@@BBB.LOADLIB
//         DD  DDNAME=SLIB2                     ===>>   DB2 LIBRARY
//*
//SLIB2    DD  DISP=SHR,DSN=DSN2410.SDSNLOAD      ===>>   DB2 LIBRARY
//*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//*
//TR#SQL   DD  DUMMY  OR SYSOUT=* SQL TRACE PRINT (FOR DEBUGGING)
//*
//SYSPUNCH DD  SYSOUT=*,DCB=(LRECL=81,BLKSIZE=81)
//* ABNLIGNR DD  DUMMY               ABEND-AID
```

Testing VISION:Report Interface to DB2

Once the interface modules have been installed, VISION:Report Interface to DB2 can be tested using the QJTSTDB2 procedure (see QJTSTDB2). Make certain that the QJTSTDB2 JCL procedure is available to the test jobs by verifying that QJTSTDB2 is in a system-accessible procedure library or by including a JOBPROC JCL statement pointing to the library that contains QJTSTDB2. QJTSTDB2 contains the required VISION:Report JCL to execute VISION:Report Interface to DB2 within a set of test programs.

The member names, testing sequence, and expected output are in the VISION:Report Interface to DB2 Reference Guide. These tests require that module QUIKDATT be present in your VISION:Report load library. If it is not present, you will need to run ASMYDATT.

Using VISION:Report Interface to DB2 Under DSN

As distributed, VISION:Report Interface to DB2 uses an internal attachment to DB2 (using the Call Attach Facility). However, your installation may require the use of the TSO Attach Facility (DSN). To use DSN, set the system name in the QJOPTION macro to all blanks. Replace the statement that executes VISION:Report in your TSOCLIST with the following statement:

```
DSN RUN PROGRAM(QUIKJOB) PLAN(QJ#PLAN) END
```

This assumes that the DB2 load library (where the DSN resides) is already allocated as part of your JOBLIB or STEPLIB (or is in the LPA), and that the VISION:Report and VISION:Report Interface to DB2 load libraries are also allocated.

Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at esupport.ca.com. Technical support is available 24 hours a day, 7 days a week.

Chapter

2 VSE Installation and Use

VISION:Report Interface to DB2 requires three components: the SQL application interface, the error message formatter and a phase consisting of one or more tables containing all the information needed to execute SQL requests from VISION:Report applications. The first two components are created from objects supplied with DB2. The last component is part of VISION:Report. The tables are created in a sequence of jobs described in the following sections. The information in each table is derived from the output of the pre-processing of a collection of model SQL statements and is saved as an object. The jobs generate one object and one package for each application server. The objects are then linked into a single phase.

The member, @SQLINDX, contains information about the installation sequence, as well as sample jobs. Please be sure to review this member before you start the installation process.

Punch the SQLINSTL.A member located in the library created by the QJINSTL job. Run the SQLINSTL job.

```
**$$ JOB JNM=SQLINSTL,CLASS=A,LDEST=(,USERID)
**$$ LST CLASS=A,DEST=(,USERID)
**$$ PUN CLASS=A,DEST=(,USERID)
// JOB SQLINSTL  INSTALL OPTIONAL SQL/DS INTERFACE
* STEP1: RESTORE SQL/DS INTERFACE LIBRARY
// OPTION LOG,NODUMP
/*
/* PLEASE DO A "MASS CHANGE" TO THE FOLLOWING
/* JCL TO FIT YOUR COMPANY'S STANDARDS (WHERE APPLICABLE):
/*        ',USERID' TO: IF UNDER VM, YOUR CMS USERID
/*        '**$$'    TO: '* $$' IF UNDER VM
/*        '@@@VLL'  TO: VOLUME SERIAL NUMBER OF QJ   LIBRARY
/*        '@@@AAA.@@@BBB.LIB' TO:   QJ LIBRARY NAME
/*        '@@TAPE'  TO: ORIGINAL DISTRIBUTION TAPE DRIVE
/*
// DLBL   QJLIB,'@@@AAA.@@@BBB.LIB',99/366    QJ   LIBRARY
// EXTENT       ,@@@VLL
// LIBDEF  PHASE,SEARCH=(QJLIB.QJ161),                                 X
               CATALOG=QJLIB.QJ161
// LIBDEF SOURCE,SEARCH=QJLIB.QJ161
// LIBDEF    OBJ,SEARCH=QJLIB.QJ161
// PAUSE  MOUNT DISTRIBUTION TAPE ON @@TAPE
// MTC    REW,@@TAPE               ENSURE TAPE AT LOAD POINT
// MTC    FSF,@@TAPE,10            SKIP TO 4TH FILE
// ASSGN SYSIPT,@@TAPE
// EXEC   LIBR,SIZE=512K,PARM='ACC SUB=QJLIB.QJ161'
/*
/&
```

```
**$$ EOJ
```

After you run this job, you have the remaining members necessary to complete the installation of VISION:Interface for DB2 in your library.

Punch the following members and modify according to your company's standards:

- LINKSQL.A
- PRESQLJ.A
- PREPROC.A
- POSTSQL.A
- ASMPOST.A
- CATLPOST.A
- LINKPOST.A

Run the following job:

- LINKSQL

Run the following jobs in the order listed below:

- PRESQLJ
- PREPROC
- POSTSQL
- ASMPOST
- CATLPOST

Repeat these steps for each application server. Run the following job after all the application request table objects have been created:

- LINKPOST

The individual jobs are described in the subsequent sections.

LINKSQL

This job creates the SQL application interface and the error message formatter phases.

```
// JOB LINKSQL
/* LINK SQL INTERFACE AND MESSAGE FORMATTER PHASES
/* CHANGE:
/* '@QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
/* '@SQLL' AND '@SQLS' TO DB2 LIBRARY AND SUBLIBRARY NAMES
/* '@SQLN' TO DB2 LIBRARY NAME
/* '@SQLVOL'  TO DB2 LIBRARY VOLSER
/*
// DLBL  @SQLL,'@SQLN'       DB2 LIB
// EXTENT ,@SQLVOL
// DLBL QJ,'@QJLIB'
// EXTENT ,@VOLSER
// LIBDEF  *,SEARCH=@SQLL.@SQLS,CATALOG=QJ.QJ161
// OPTION CATAL
   PHASE ARIPRDID,S
    INCLUDE ARIPRDID
   PHASE ARIMSMF,S
    INCLUDE ARIMSMF
    INCLUDE ARIMMGM
    INCLUDE ARISYSDD
/*
// EXEC LNKEDT
/*
/&
```

Creating an Application Request Table

There are five jobs that create one application request table object. The first job, PRESQLJ, generates the model SQL statements. The second job, PREPROC, preprocesses the SQL statements. The third job, POSTSQL, creates an assembler language source from the preprocessed SQL statements. The fourth job, ASMPOST, assembles the source into an object. The fifth job, CATLPOST, catalogs the object into your VISION:Report library.

PRESQLJ

This job generates the model SQL statements.

A VISION:Report Interface to DB2 application request table is 400 bytes in size plus 20 bytes for each cursor generated. You can generate the table with entries for 1 to 999 cursors.

Insert values into the following parameters of the PRESQL macro:

VERSION= and RELEASE= **--** used to control the generation of SQL statements to be used as input to the SQL preprocessor and to control the generation of the parameter list to the SQL application interface.

AUTHNAME= and PASSWORD= **--** used to execute the initial connect to the default application server. Connect authority must be granted to the authorization id with the password specified here and to any authorization id with the password specified in a SQL CONNECT statement from within a VISION:Report program. If everyone at your installation needs this authority, you may grant connect to all users.

APPLID= and SERVER= **--** used to point to the correct application request table following the execution of an explicit CONNECT statement.

CURSORS= **--** used to determine the number of cursors that will be generated. Set this value according to the number of cursors that you will anticipate needing for your applications.

```
// JOB PRESQLJ
/* GENERATE MODEL SQL STATEMENTS
/* CHANGE:
/* '@QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
// DLBL QJ,'@QJLIB'
// EXTENT ,@VOLSER
// LIBDEF  *,SEARCH=QJ.QJ161
// DLBL    IJSYSPH,'QJ.SQL1',1,SD
// EXTENT SYSPCH,@VOLSER,,,1,100
    ASSGN  SYSPCH,DISK,VOL=@VOLSER,SHR
// OPTION DECK
// EXEC    ASSEMBLY
         PRESQL VERSION=6,RELEASE=1,CURSORS=08,                        X
               AUTHNAME=DYLA,PASSWORD=DYLAPW,APPLID=SYSARI00,          X
               SERVER=SQLDS
         END
/*
/&
    CLOSE  SYSPCH,PUNCH
```

Following are sample control statements for two additional application servers:

```
         PRESQL VERSION=6,RELEASE=1,CURSORS=12,                        X
               AUTHNAME=DYLB,PASSWORD=DYLBPW,APPLID=SYSARI01,          X
               SERVER=DB261DB

         PRESQL VERSION=3,RELEASE=5,CURSORS=18,                        X
               AUTHNAME=DYLC,PASSWORD=DYLCPW,APPLID=SYSARI02,          X
               SERVER=SQL35DB
```

PREPROC

This job preprocesses the model SQL statements. USER= is the user id with the password granted DBA authority by the database administrator to install VISION:Interface for DB2. PREP= is the name of the package on the server. DBNAME is the name of the application server.

```
// JOB    PREPROC
/* PREPROCESS MODEL SQL STATEMENTS
/* CHANGE:
/* '@GLOB' TO SQL GLOBAL VARIABLE FILE NAME
/* '@SQLCAT' TO VSAM CATALOG ID
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
/* '@SQLL' AND '@SQLS' TO DB2 LIBRARY AND SUBLIBRARY NAMES
/* '@SQLN' TO DB2 LIBRARY NAME
/* '@SQLVOL' TO DB2 LIBRARY VOLSER
// DLBL   @SQLL,'@SQLN'               DB2 LIB
// EXTENT ,@SQLVOL
// DLBL   SQLCAT,'SQL.CATALOG',,VSAM
// DLBL   SQLGLOB,'@GLOB',,VSAM,CAT=@SQLCAT,DISP=(OLD,KEEP)
// LIBDEF  *,SEARCH=@SQLL.@SQLS
// DLBL   IJSYSIN,'QJ.SQL1'
// EXTENT SYSIPT,@VOLSER
   ASSGN  SYSIPT,DISK,VOL=@VOLSER,SHR
// DLBL   IJSYSPH,'QJ.SQL2',1,SD
// EXTENT SYSPCH,@VOLSER,,,1,200
   ASSGN  SYSPCH,DISK,VOL=@VOLSER,SHR
// EXEC   PGM=ARIPRPA,SIZE=AUTO,PARM='USER=SQLDBA/SQLDBAPW,PREP=QJ0001,X
               DBNAME=SQLDS'
/*
/&
   CLOSE  SYSPCH,PUNCH
   CLOSE  SYSIPT,READER
```

Following are sample EXEC cards for two additional application servers:

```
// EXEC   PGM=ARIPRPA,SIZE=AUTO,PARM='USER=SQLDBA/SQLDBAPW,PREP=QJ0002,X
               DBNAME=DB261DB'
// EXEC   PGM=ARIPRPA,SIZE=AUTO,PARM='USER=SQLDBA/SQLDBAPW,PREP=QJ0003,X
               DBNAME=SQL35DB'
```

POSTSQL

This job creates an assembler language source for an application request table from the preprocessed SQL statements.

```
// JOB    POSTPROC
/* GENERATE ASSEMBLER LANGUAGE SOURCE FOR APPLICATION REQUEST TABLE
/* CHANGE:
/* '@QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
// DLBL QJ,'@QJLIB'
// EXTENT ,@VOLSER
// LIBDEF  *,SEARCH=QJ.QJ161
// DLBL   IJSYSIN,'QJ.SQL2'
// EXTENT SYSIPT,@VOLSER
   ASSGN  SYSIPT,DISK,VOL=@VOLSER,SHR
// DLBL   IJSYSPH,'QJ.SQL3',1,SD
// EXTENT SYSPCH,@VOLSER,,,1,100
   ASSGN  SYSPCH,DISK,VOL=@VOLSER,SHR
// EXEC   PGM=POSTSQL,SIZE=AUTO
/*
/&
   CLOSE  SYSPCH,PUNCH
   CLOSE  SYSIPT,READER
```

ASMPOST

This job assembles the source into an object.

```
// JOB    ASMPOST
/* ASSEMBLE APPLICATION REQUEST TABLE
/* CHANGE:
/* 'QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
// DLBL   QJ,'@QJLIB'
// EXTENT ,@VOLSER
// DLBL   IJSYSIN,'QJ.SQL3'
// EXTENT SYSIPT,@VOLSER
   ASSGN  SYSIPT,DISK,VOL=@VOLSER,SHR
// DLBL   IJSYSPH,'QJ.SQL4',1,SD
// EXTENT SYSPCH,@VOLSER,,,1,100
   ASSGN  SYSPCH,DISK,VOL=@VOLSER,SHR
// OPTION DECK
// EXEC   ASSEMBLY
/*
/&
   CLOSE  SYSPCH,PUNCH
   CLOSE  SYSIPT,READER
```

CATLPOST

This job catalogs the object into your VISION:Report library.

```
// JOB    CATLPOST
/* CATALOG APPLICATION REQUEST TABLE
/* CHANGE:
/* '@QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
// DLBL QJ,'@QJLIB'
// EXTENT ,@VOLSER
// DLBL   IJSYSIN,'QJ.SQL4'
// EXTENT SYSIPT,@VOLSER
   ASSGN  SYSIPT,DISK,VOL=@VOLSER,SHR
// EXEC   PGM=LIBR,SIZE=256K,PARM='ACCESS SUBLIB=QJ.QJ161;CATALOG SQLOB-
               J01.OBJ REPLACE=YES'
/*
/&
   CLOSE  SYSIPT,READER
```

Following are sample EXEC cards for two additional application servers:

```
// EXEC   PGM=LIBR,SIZE=256K,PARM='ACCESS SUBLIB=QJ.QJ161;CATALOG SQLOB-
               J02.OBJ REPLACE=YES'
// EXEC   PGM=LIBR,SIZE=256K,PARM='ACCESS SUBLIB=QJ.QJ161;CATALOG SQLOB-
               J03.OBJ REPLACE=YES'
```

The names assigned to the objects can be anything you choose. The object names are not used by VISION:Report. Use any names meaningful to yourself.

Creating an Application Request Phase

The application request phase is created from the application request table objects created in the previous steps. The object created for your default server must be ordered first. This is because VISION:Report performs a CONNECT to your default server at the beginning of program execution using the authorization id and the password specified in the object. The objects for the remaining application servers may be in any order.

LINKPOST

This job links the object or objects created in the previous steps into a single phase in your VISION:Report library. There is one INCLUDE statement for each object that you have created in the preceding jobs. Order the object for your default server first.

```
// JOB LINKPOST
/* GENERATE APPLICATION REQUEST PHASE
/* CHANGE:
/* '@QJLIB' TO QJ LIBRARY NAME
/* '@VOLSER' TO VOLUME SERIAL NUMBER OF QJ LIBRARY
// DLBL QJ,'@QJLIB'
// EXTENT ,@VOLSER
// LIBDEF *,SEARCH=QJ.QJ161,CATALOG=QJ.QJ161
// OPTION CATAL
 PHASE SQLCSECT,S
 INCLUDE SQLOBJ01
 INCLUDE SQLOBJ02
 INCLUDE SQLOBJ03
/*
// EXEC LNKEDT
/*
/&
```

Customizing the VISION:Report QJOPTION Macro

When VISION:Report Interface to DB2 is run with VISION:Report, the following keywords should be noted in the assembly of the QJOPTION macro: SQLA1-SQLA5. Refer to the VISION:Report Reference Guide, OPTION verb.

Sample Jobs

Once the installation process has been completed, you may test the system by modifying the various jobs that are listed in the member, @SQLINDX and executing them in the listed order.

Index