

IBM DB2 Bind Manager for z/OS
Version 2 Release 4

User's Guide



IBM DB2 Bind Manager for z/OS
Version 2 Release 4

User's Guide



Note:

Before using this information and the product it supports, read the information in "Notices" on page 131.

This edition applies to Version 2 Release 4 of IBM DB2 Bind Manager for z/OS (product number 5655-E43) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC19-2751-03.

© **Copyright IBM Corporation 2000, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information	v
Service updates and support information	v
Highlighting conventions	v
How to look up message explanations	vi
Searching an information center.	vi
Using a Web search.	vi
Using LookAt.	vi
How to send your comments	vii
Figures	ix
Tables	xi
Chapter 1. DB2 Bind Manager overview	1
What does DB2 Bind Manager do?	2
DB2 Bind Manager features and benefits	2
DB2 Bind Manager architecture and process flow	4
DB2 Bind Manager and DB2 Path Checker scenario	6
DB2 Bind Manager publications and updates	7
Accessibility features	8
Summary of changes	8
Chapter 2. Configuration	13
Environment support	13
Requirements.	13
Configuration summary	14
Configuration steps.	14
Step 1: Bind the DB2 Bind Manager plans	14
Step 2: Change the module name	18
Step 3: Configure the JCL for your installation.	18
Step 4: Configure DB2 Bind Manager for DB2I.	20
Step 5: Copy the BNDAVB module to an authorized library	20
Accessing DB2 Bind Manager from the DB2 Admin Launchpad	21
Chapter 3. DBRM and SQL analysis tools	23
Using DBRM load module cross-reference	23
Running ISPF mode	23
DBRM batch cross-reference (program BNDA006)	36
Creating the static SQL load module report.	41
Catalog cleaning (program BNDA020)	42
BNDA020 job control statements (JCL)	42
BNDA020 commands	43
DBRM creation (program BNDA024)	45
BNDA024 job control statements (JCL)	45
BNDA024 commands	46
Catalog timestamp listing (program BNDA030)	48
BNDA030 job control statements (JCL)	48
BNDA030 commands	49
Catalog cleanup (program BNDA036)	50
BNDA036 job control statements	51
BNDA036 commands	52
Dynamic Mode for BNDA036	55
The Checkpoint/Restart facility.	58
Creating the index for BNDA036	59

BNDA055	61
Running BNDA055	61
BNDA055 input and output	62
BNDA055 job control statements	62
Sample BNDA055 job	63
Batch DBRM interpreter (BNDA043)	64
BNDA043 job control statements	64
BNDA043 commands	65
Example BNDA043 output report	66
Catalog cleanup (program BNDA046)	67
BNDA046 job control statements	67
BNDA046 commands	68
Creating the index for BNDA046	69
Chapter 4. Bind Avoidance	73
Using BNDAVB	74
Configuring coprocessor mode (program BNDAVB4)	74
Using BNDAVB3 to eliminate BIND processing after a precompile	78
Generating SMF records	79
Changing the report destination	80
Checking for precompiler warnings	80
Activating DBRM generation	81
Ignoring DECLARE TABLE statements	81
Suppressing CCSID checking (NOCC)	81
Chapter 5. Troubleshooting	83
Gathering diagnostic information	83
DB2 Bind Manager Messages and codes	84
DB2 Bind Manager Return codes	84
DB2 Bind Manager Message severities	85
DB2 Bind Manager messages	86
Notices	131
Trademarks	132
Bibliography	133
Index	135

About this information

IBM DB2® Bind Manager for z/OS (also referred to as DB2 Bind Manager) allows application programmers to safely bypass the DB2 bind process for code changes that do not alter existing SQL structures in an application. The DBRM Checker feature provides additional system integrity by identifying individual plans in a package that need to be rebound. The DBRM Load Module Cross-Reference report identifies the DB2 token in a DBRM library to the DB2 subsystem and to CSECTs within specified load modules. The static SQL report shows which load modules and CSECTs contain static SQL.

These topics provide instructions for using IBM® DB2 Bind Manager and DBRM Checker for z/OS®.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:

- Plan for the installation of DB2 Bind Manager
- Install and operate DB2 Bind Manager
- Customize your DB2 Bind Manager environment
- Diagnose and recover from DB2 Bind Manager problems
- Design and write applications for DB2 Bind Manager

To use these topics, you should have a working knowledge of:

- The OS/390® or z/OS operating system
- ISPF
- SMP/E

Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Always check the DB2 Tools Product publications page for the most current version of this publication:

<http://www.ibm.com/software/data/db2imstools/db2tools-library.html>

Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

www.ibm.com/software/data/db2imstools/support.html

Highlighting conventions

This information uses the following highlighting conventions:

- **Boldface** type indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- **Monospace** type indicates examples of text that you enter exactly as shown.

- *Italic* type indicates variables that you should replace with a value, to indicate the titles of other publication, and to emphasize significant terms.

How to look up message explanations

You can use any of the following methods to search for messages and codes:

Searching an information center

In the search box that is located in the top left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center, enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to help you improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use wild cards (* or ?) in the message number to broaden your search; for example, DFS20??.I.

The information center contains the latest message information for all of the information management products that are included in the information center.

Using a Web search

You can use any of the popular search engines that are available on the Web to search for message explanations. When you type the specific message number or code into the search engine, you will be presented with links to the message information in IBM information centers.

Using LookAt

LookAt is an online facility that you can use to look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM[®], VSE/ESA, and Clusters for AIX[®] and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OSe systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS).
- Your Microsoft Windows workstation. You can install code to access IBM message explanations on the z/OS Collection (SK3T-4269) using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your z/OS Collection (SK3T-4269) or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DB2 Bind Manager documentation, use either of the following options:

- Use the online reader comment form, which is located at:
www.ibm.com/software/data/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of DB2/IMS Product_name, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Figures

1.	DB2 application compile process	5
2.	DBRM Checker	5
3.	JCL sample job BNDBIND	18
4.	Sample JCL for using DB2 Bind Manager	20
5.	CLIST BNDC002 code modification	25
6.	DB2 Bind Manager main menu panel	26
7.	DB2 Bind Manager - SETTINGS panel	26
8.	DBRM Cross Check panel.	27
9.	DBRM module cross-reference selection list	27
10.	DBRM module cross-reference results	27
11.	Package Cross Check panel	28
12.	Sample DBRM verification results	29
13.	Sample DBRM verification results for a specific instance panel	29
14.	Browse DBRM panel	30
15.	Sample DBRM selection panel	30
16.	DBRM output.	31
17.	The DBRM Rebuilder panel	32
18.	Plan/DBRM selection panel	32
19.	Sample Plan/DBRM selection panel	33
20.	Specifying the output library.	33
21.	DBRM created	33
22.	Inspecting the re-created DBRM.	34
23.	The DB2 Bind Manager Index Utility panel	34
24.	Available Indexes panel	35
25.	Available indexes with a selected index	35
26.	Index key columns panel	35
27.	Sample JCL for BNDA006.	37
28.	BNDA006 sample cross-reference report	40
29.	Sample JCL for static SQL load module report	42
30.	Static SQL load modules	42
31.	Sample job using the Dynamic Mode features (1 of 2)	57
32.	Sample job using the Dynamic Mode features (2 of 2)	58
33.	Sample member BNDJ36I JCL (1 of 2).	60
34.	Sample member BNDJ36I JCL (2 of 2).	61
35.	Sample BNDA055 job	64
36.	JCL for sample BNDA043 output report	64
37.	Sample member BNDJ46I JCL (1 of 2).	70
38.	Sample member BNDJ46I JCL (2 of 2).	71
39.	Coprocessor support sample JCL (Part 1).	77
40.	Coprocessor support sample JCL (Part 2).	78

Tables

1.	Overview of configuration steps	14
2.	Job control statements for BNDA006	36
3.	Job control statements for BNDA020	43
4.	Job control statements for BNDA024	45
5.	DEFAULTS command values.	47
6.	Job control statements for BNDA030	49
7.	Job control statements for BNDA036	51
8.	Job control statements for BNDA055	62
9.	Job control statements for BNDA043	65
10.	Job control statements for BNDA046	68
11.	Job control statements for BNDAVB4	74
12.	Job control statements for BNDAVB1	75
13.	Job control statements for BNDAVB3	79
14.	DB2 Bind Manager SMF record	80
15.	DB2 Bind Manager return codes	84
16.	DBRM Cross-Reference return codes	84
17.	DBRM Checker return codes	84
18.	BNDAVB3 return codes	85
19.	BNDAVB4 return codes	85
20.	DBRM Checker message severities	85
21.	BND635E return codes	104
22.	BND635E return code actions	104
23.	BND636I return codes	104

Chapter 1. DB2 Bind Manager overview

IBM DB2 Bind Manager for z/OS (also referred to as DB2 Bind Manager) streamlines the development, testing, and implementation of DB2 applications by alleviating one of the principal bottlenecks affecting these processes—the DB2 bind.

Preparing a DB2 application for processing involves several steps. The particular combination and sequence of these steps depend on the environment where the application is developed and the environment where it will run. However, nearly all applications require some form of compilation or language translation as well as linkage editing to create an executable load module. Some applications require further preparation steps that are unique to their particular environments, such as a CICS® new copy function. Precompilation and binding are two directly-related preparation steps that are specific to DB2.

Before you can compile a program using embedded (static) SQL, it must first be processed by the DB2 precompiler. The precompiler performs several essential functions:

- It performs extensive validation of the SQL statements. This includes verifying syntax and structure, and matching host language variables and their attributes to the SQL statements that reference them.
- It replaces SQL statements with appropriate host language statements, usually some form of subroutine call. It also inserts a timestamp value into the source program. This timestamp is used by DB2 integrity checking.
- It extracts the SQL statements and stores them in a condensed form as a Database Request Module (DBRM).

After a program with embedded SQL is successfully processed by the DB2 precompiler and the appropriate host language compiler, it must undergo a DB2 bind before it is issued. Binding establishes a fixed relationship between a program and the databases it will access.

The bind process is complex and involves selecting access paths to the data, further SQL verification, and various other checks and validations. Until a program is successfully bound, it cannot access DB2 data.

The primary input to the bind process is the DBRM that was created when the program was processed by the precompiler. During the bind process, DB2 uses information in the DBRM to create a valid connection between the application program and the DB2 databases it will access.

The result of the bind is an application plan or package, which is stored in the DB2 catalog. The plan or package contains a timestamp value that is compared at the time it is issued to the one existing in the program. If the two do not agree, the program fails with a SQL return code of -818 or -805. This timestamp validation process causes applications to be rebound after every compilation, whether the SQL statements were modified or not.

Binding should be avoided whenever possible. The need to bind applications before using them requires time and valuable system resources.

As an application platform and enterprise data repository, DB2 becomes increasingly mission-critical. Effective and efficient use of DB2 is often vital to an enterprise's success. This means not only that the online operation of the database must be robust and reliable, but also that the process of developing and implementing applications be made as efficient and free of roadblocks as possible. It is this latter requirement that DB2 Bind Manager addresses.

Topics:

- "What does DB2 Bind Manager do?"
- "DB2 Bind Manager features and benefits"
- "DB2 Bind Manager architecture and process flow" on page 4
- "DB2 Bind Manager and DB2 Path Checker scenario" on page 6
- "DB2 Bind Manager publications and updates" on page 7
- "Accessibility features" on page 8
- "Summary of changes" on page 8

What does DB2 Bind Manager do?

DB2 Bind Manager allows you to modify programs without involving DB2. It lets you safely bypass the DB2 bind process for source code changes that do not alter the existing SQL structure.

The DBRM Checker feature of DB2 Bind Manager helps maintain DB2 system integrity. This function enables you to determine whether a bind is required by comparing the consistency tokens for a plan in a DBRMLIB with the corresponding tokens in the DB2 catalog tables. Because DB2 Bind Manager does not connect to DB2 to check DBRM validity, DBRM Checker can also be used to insure that the correct DBRMLIB is accessed during the compile process.

DBRM load module cross-reference allows for validation of the DB2 token in the load module. This validation is performed on a DBRM to load module basis through an ISPF screen. The same function is available in batch for comparisons between members of DBRM libraries and load libraries. For selection in batch mode, you can use wild cards.

DB2 Bind Manager features and benefits

DB2 Bind Manager helps in the efficient development of DB2 applications by eliminating unnecessary binds, saving time, and conserving DB2-related resources.

DB2 Bind Manager runs as part of a normal DB2 application development process in batch or under DB2I. DB2 Bind Manager automatically analyzes each precompile to identify changes in the SQL structure and determine if a bind is actually needed. DB2 Bind Manager only performs required binds. In other words, it never performs a DB2 bind that is not completely necessary. With DB2 Bind Manager, you will never do another unnecessary DB2 bind. Depending on the type of maintenance activity, DB2 Bind Manager can eliminate up to 50 percent or more of the binds in your installation. DB2 Bind Manager can help to resolve the following problems:

- **Binding can be a procedural bottleneck**

The bind process can be a costly bottleneck. In most cases, DB2 must completely quiesce access to a plan for that plan to be rebound. In busy online transaction processing environments such as IMS™ or CICS, binding might be difficult or

impossible during normal business operations. As a result, newly updated applications cannot be moved to production when needed.

- **Binding lowers productivity**

If programmers must wait hours (or even overnight) to transfer updated applications to test or production, their productivity suffers. End-user productivity is impacted when modified applications are unavailable, and business opportunities can be lost when an application or online system must be shut down to make an emergency change.

- **Rebinding can affect performance**

Whenever you rebind a program, DB2 reevaluates the access paths that DB2 Bind Manager uses to read or write the data. Often, reevaluating the access paths results in different paths being selected. Depending on changes made to the databases since the initial bind, these new paths might not be the most efficient for a specific query. Sometimes, the first indication that an access path has changed occurs when an application's response time worsens for no apparent reason. (You can test for potential access path changes with IBM DB2 Path Checker, Product Number 5655-E39.)

DB2 Bind Manager also benefits change management. When integrated into a production promotion scheme, DB2 Bind Manager will automatically detect any production application change that requires a bind. DBA's can then concentrate their efforts on only those changes that affect the SQL structure.

DB2 Bind Manager also includes the following options:

DBRM Generation

DB2 Bind Manager compares the results of the current precompile (the *new* DBRM) to the results of the previous precompile (the *old* DBRM). This comparison requires that the old DBRM is available when DB2 Bind Manager is processing.

However, if it is not the policy at your installation to save DBRMs, the old DBRM is not available for comparison. DBRM Generation circumvents this limitation by generating a temporary copy of the old DBRM from information in the DB2 catalog which allows the comparison to proceed normally. The temporary DBRM is deleted when the job step ends.

Activate DBRM Generation by adding a DD statement to the execution JCL. Preface the ddname with SSID to signal to DB2 Bind Manager that DBRM Generation is in effect. The remaining characters of the ddname (one to four) specifies the DB2 subsystem name where DBRM information will be extracted. In the following example, DBRM generation is invoked using subsystem DSN1.

```
//SSIDDSN1 DD DUMMY
```

DECLARE TABLE exception

Invoking the DECLARE TABLE exception causes DB2 Bind Manager to ignore DECLARE TABLE statements when comparing old and new DBRMs.

Activate the DECLARE TABLE exception by adding a special DD statement to the execution JCL. The statement ddname must be SKIPDECL as shown in the following example:

```
//SKIPDECL DD DUMMY
```

Coprocessor support

In a traditional operating environment, the DB2 precompiler runs as a separate job step. DB2 Bind Manager intercepts the source code after it is

modified by the precompiler and before the language compiler sees it. DB2 Bind Manager operates on this intermediate source code before passing it on to the compiler.

In a coprocessor environment, the precompiler and the language compiler run together in a single step. Thus, the intermediate source code is not available for manipulation. With DB2 Bind Manager coprocessor support, DB2 Bind Manager functionality also operates in the coprocessor environment so that bind avoidance continues to be fully supported.

Bind Avoidance

When migrating between DB2 Version 7 and DB2 Version 8 compatibility mode, DB2 Version 7 inserts a CCSID into the DBRM. DB2 Bind Manager monitors the operation of the DB2 precompiler and determines if a bind is required by comparing the new and old DBRMs.

DB2 Bind Manager will compare the incoming DBRM to the DBRM generated from the targeted DB2 subsystem where the bind might occur. The result is either a bind or a bind avoidance. Then, if the DBRM is passed to the target subsystem and the incoming DBRM is compared to the generated DBRM, the connect token in the DBRM is either matched or not matched to the DB2 catalog. If there is a match, the bind is avoided. If there is not a match, the bind is not avoided.

Bind Avoidance does not connect to an active DB2 subsystem to validate DBRM and plan information except when the optional DBRM Generation feature is used.

This feature is offered as an option; it is not a default setting. When data is moved between EBCDIC and Unicode, a bind is always required.

DB2 Bind Manager architecture and process flow

DB2 Bind Manager monitors the DB2 precompiler processing and, by analyzing both the new and old DBRMs, determines whether a bind is required. DBRM Checker performs consistency checks between an existing DBRMLIB and a DB2 subsystem.

DB2 Bind Manager must have access to the library that contains the DBRMs that were used during the previous bind operation. It assumes that the information in this DBRM library is identical to that used by DB2. DB2 Bind Manager does not connect to any active DB2 subsystem to validate DBRM and plan information except when DBRM Generation is used.

The flowchart in the following figure illustrates the program preparation process with DB2 Bind Manager installed and operating. The area inside the dotted lines depicts the part of the process that DB2 Bind Manager affects.

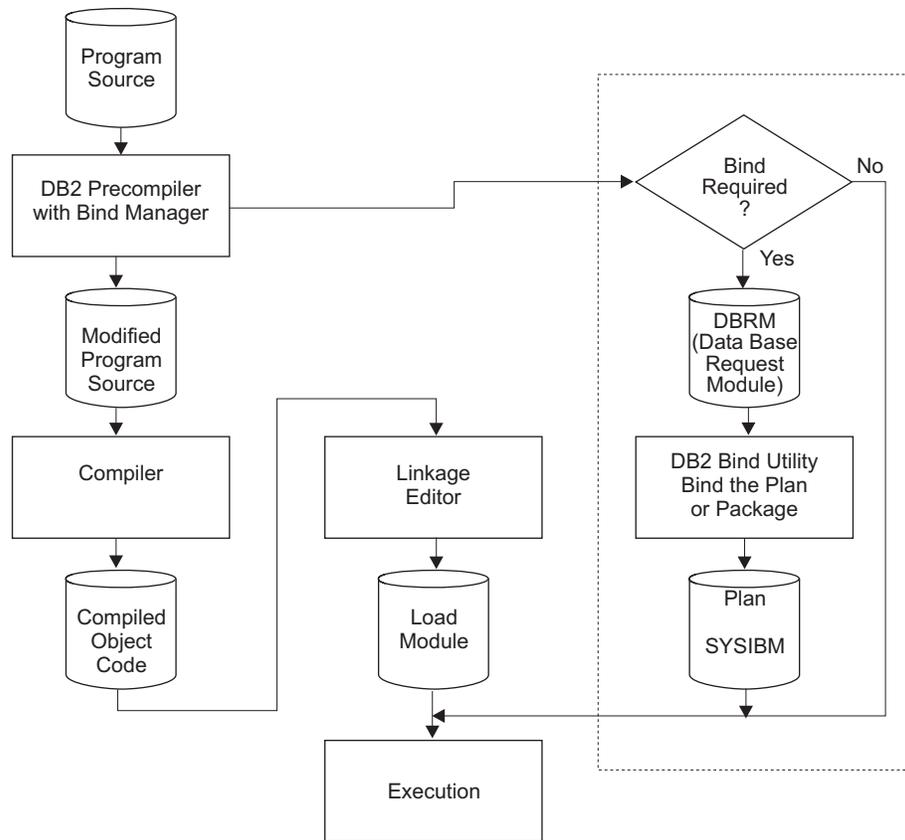


Figure 1. DB2 application compile process

DBRM Checker

DBRM Checker identifies DBRMs by plan that have consistency tokens that do not match those in the DB2 catalog tables. DBRM Checker creates a report that identifies any mismatches. The following diagram illustrates this process.

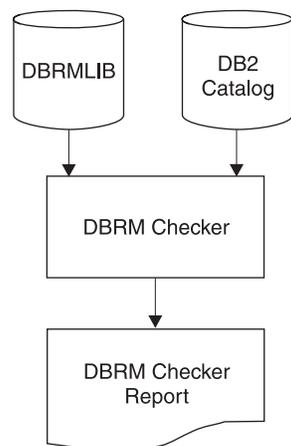


Figure 2. DBRM Checker

DBRM Checker does not require a bind to perform its analysis.

DB2 tokens in load modules can be compared with DBRMs and the DB2 subsystem. This comparison can be performed through ISPF for a single DBRM. For comparisons of large numbers of DBRMs to large numbers of load modules, a batch program is used. DBRM Checker can also report load modules and the CSECTs within the load modules that contain the static SQL.

DB2 Bind Manager and DB2 Path Checker scenario

You can use DB2 Bind Manager to address many of your day-to-day business problems.

The following scenario illustrates how you can use DB2 Bind Manager and DB2 Path Checker to address typical business problems.

Tool efficiency to reduce costs

Two tools from IBM help you avoid costly and unexpected results from bind operations. IBM DB2 Path Checker for z/OS enables you to identify, analyze, and document in advance potential access path changes without initiating bind processing. DB2 Bind Manager automatically eliminates unnecessary binds and processes only necessary DB2 binds.

Smoother migration to DB2 for z/OS

When migrating to a new version of DB2 for z/OS, these tools greatly reduce the amount of catalog data to be migrated. They also allow you to proactively identify potentially harmful access path changes before they occur. New features of the tools include:

- **DB2 Path Checker:** Use sample SPUFI queries before a migration to automatically populate PLAN_TABLES with the information needed to identify possible high-risk access path changes.
- **DB2 Bind Manager:** Use the “Catalog cleanup” feature to compare bound packages in a DB2 subsystem to one or more load libraries and generate FREE commands to remove obsolete or unreferenced packages from the catalog

DB2 Path Checker

Using DB2 Path Checker, you can quickly determine whether a bind of a database request module (DBRM) will result in a changed access path. You can also see the potential effects of doing a bind on one or many programs. DB2 Path Checker initiates an EXPLAIN into the plan table of the new DBRM and gives you a report on paths that have changed, or on all paths.

Using DB2 Path Checker, you can:

- Obtain an EXPLAIN on a standalone DBRM to see which paths will change, or use Lists and Pattern Matching to run a path check on a batch of DBRMs in one pass.
- Compare batches of DBRMs in one subsystem against another subsystem after doing a bind to identify potential problems.
- Compare two EXPLAINS based on the types of SQL used and the tables accessed using matching logic (MATCHSQL). This comparison allows SQL to change position in the program and in relation to other SQL and still be matched for comparison.
- Find the DBRM that has no matching EXPLAIN.

- Use wildcards for plans and packages for both comparison and reporting.
- Use the TEST Command to determine what will change when migrating between subsystems and which DBRMs will have issues if rebound.

DB2 Bind Manager

After a program with embedded SQL has been successfully processed by the DB2 precompiler, DB2 Bind Manager analyzes each precompile to determine whether the SQL structure has changed. If not, then a bind is not performed. Since DB2 Bind Manager detects only production application changes requiring a bind, it frees database administrators from doing unnecessary binds and allows them to concentrate on changes that affect the SQL structure.

DB2 Bind Manager offers these features:

- An interactive system productivity facility (ISPF) interface enables you to select individual DBRMs from the DBRM library and compare them against the subsystem and the load module to verify that you are applying the correct program. If you have missing DBRMs, you can recreate them from the system catalog.
- Using Lists and Pattern Matching, you can compare batches of DBRMs against subsystems and load modules in one pass.
- DBRM Checker can go through a load module and determine which programs and which modules within those programs contain static SQL.
- A program does mass comparisons between a DB2 subsystem and different load libraries. Where there is no matching load module, a FREE command for the package is created and a backup of the DBRM is generated. This action reduces the amount of catalog data that has to be converted and DBRMs needing to be rebound as part of the conversion.

DB2 Bind Manager publications and updates

This topic explains where to find DB2 Tools information on the Web, and explains how to receive information updates automatically.

DB2 Bind Manager information on the Web

The DB2 Tools Product publications Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

<http://www.ibm.com/software/data/db2imstools/db2tools-library.html>

IBM Redbooks® publications that cover DB2 and IMS Tools are available from the following Web page:

<http://www.ibm.com/software/data/db2imstools/db2tools-library.html>

Receiving updates automatically

To automatically receive a weekly email that notifies you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Support service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Support page is displayed, click **add products** to select those products that you want to receive information updates about. The DB2 and IMS Tools category is located under **Software > Data and Information Management > Database Tools & Utilities**.
4. Click **Subscribe to email** to specify the types of updates that you would like to receive.
5. Click **Update** to save your profile.

Accessibility features

The accessibility features of DB2 Bind Manager enable users with physical impairments to use software products successfully.

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in DB2 Bind Manager enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1, SC34-4822*
 - *z/OS TSO/E Primer, SA22-7787*
 - *z/OS TSO/E User's Guide, SA22-7794*

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Summary of changes

This topic summarizes the technical changes for each edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

| **SC19-2751-04 (June 2012)**

- The BNDA055 program has been added.
- Multiple messages about the BNDA036 program and the BNDAVB3 program have been added.

| **SC19-2751-03 (December 2010)**

- The section about Bind Avoidance has been revised. See Chapter 4, "Bind Avoidance," on page 73 for more information.
- A new informational message has been added to show which DBRM is being processed.

| **SC19-2751-02 (August 2010)**

|

- New DD statements have been added to program BNDA036, the DB2 Bind Manager catalog clean-up program. See “BNDA036 job control statements” on page 51 for more information.
- New commands have been added to program BNDA036. See MAXPKG, MAXLM, and RESTART for more information.
- The Checkpoint/Restart facility has been added to help manage long BNDA036 run times when you have a large number of load modules to scan. See “The Checkpoint/Restart facility” on page 58 for more information.

SC19-2751-01 (February 2010)

- You can use the BNDAVB3 program to avoid BIND processing after a successful BNDAVB precompile. See “Using BNDAVB3 to eliminate BIND processing after a precompile” on page 78.

SC19-2751-00 (November 2009)

- The batch DBRM Interpreter, BNDA043, is a new program designed to produce the same output as the existing DBRM browser ISPF function. See “Batch DBRM interpreter (BNDA043)” on page 64 for more information.
- The BNDA006 cross-reference report has been enhanced to allow multiple DBRMLIBs as input. See “DBRM batch cross-reference (program BNDA006)” on page 36 for more information.
- Messages have been added for the batch DBRM Interpreter, BNDA043. See “DB2 Bind Manager messages” on page 86 for more information.
- Information about how you can use DB2 Path Checker and DB2 Bind Manager to improve DB2 migrations has been added. See “DB2 Bind Manager and DB2 Path Checker scenario” on page 6 for more information.

SC27-1450-05 (October 2008)

- Detailed information about the error logs, messages, and other documentation that you need before you contact IBM Software Support has been added. Gathering this information prior to reporting a problem will help IBM Software Support to solve your problem more quickly and efficiently. See “Gathering diagnostic information” on page 83 for more information.
- Additional information about the SSID command format has been added. This information clarifies the format of the SSID command for the BNDA036 and BNDA046 programs. See “BNDA036 commands” on page 52 and “BNDA046 commands” on page 68 for more information.
- The instructions for creating the indexes for the BNDA036 and BNDA046 programs have been revised. This information will help you improve the performance of these programs. See “Creating the index for BNDA036” on page 59 and “Creating the index for BNDA046” on page 69 for more information.
- If you have DB2 Administration Tool installed on your system, you can add an entry to the Launchpad table that will allow you to start DB2 Bind Manager from the DB2 Administration Tool Launchpad. See “Accessing DB2 Bind Manager from the DB2 Admin Launchpad” on page 21 for more information.
- Information about the PACKAGE command has been added to BNDA036 commands section. This information explains situations in which the BNDA036 program will generate FREE commands and lists

the programs that BNDA036 will generate FREE commands for. See “BNDA036 commands” on page 52 for more information.

- Additional information has been added to error message BND019E. This information can help prevent the error that is associated with this message from occurring in the future.

SC27-1450-03 (January 2007)

This release includes numerous documentation updates resulting from APARs and other software enhancements.

SC27-1450-02 June 2005

This section summarizes the significant improvements or enhancements for DB2 Bind Manager for z/OS, Version 2 Release 3:

Support for remote system access using DDF

This release supports remote system access using DDF. A new field to specify the remote subsystem was added to “Inspecting plans” on page 26 and “Inspecting packages” on page 27. DDF is invoked when a remote subsystem is specified in one of these fields.

Index utility

An index utility quickly determines the indexes and key columns that are available for a specific table. For more information about the index utility, refer to “Determining available indexes for tables (Index utility)” on page 34.

Catalog cleaning

A catalog cleaning feature (BNDA020) correlates the package entries in the DBRM catalog to one or more load libraries to determine which packages are still referenced by some load modules. For each DBRM entry that is not referenced by a load module, the FREE command is generated and written to a sequential data set. For more information about the catalog cleaning function, refer to “Catalog cleaning (program BNDA020)” on page 42.

DBRM Builder batch utility

The DBRM Builder batch utility (BNDA024) creates DBRMs from the catalog. This function generates multiple DBRMs in a single run, unlike the online environment that only allows you to generate one at a time. For more information about the DBRM builder batch utility, refer to “DBRM creation (program BNDA024)” on page 45.

DBRM/Catalog timestamp correlation batch report

The DBRM/Catalog timestamp correlation batch report (BNDA030) identifies all DBRMs in the catalog and identifies any DBRM timestamps that do not match their corresponding timestamp in the DBRM. With this report you can predict which instances in the catalog might have become obsolete. For more information about the timestamp report, refer to “Catalog timestamp listing (program BNDA030)” on page 48.

Suppress character set (CCSID) checking

You can opt to suppress character set (CCSID) checking when comparing DBRMs. This function causes DB2 Bind Manager to skip the CCSID check but perform all other checks as usual. Refer to “Suppressing CCSID checking (NOCC)” on page 81 for additional information.

DB2 Version 8

Support was added for new DB2 Version 8 functionality.

SC27-1450-01 (April 2004)

This section summarizes the significant improvements or enhancements for DB2 Bind Manager for z/OS, Version 2 Release 2:

DBRM Generation

DBRM Generation temporarily re-creates a missing DBRM long enough for DB2 Bind Manager to perform its analysis. This temporary DBRM is not saved anywhere; it is discarded after use.

DBRM Rebuilder

With DBRM Rebuilder (GENERATE function), you can re-create a DBRM missing from the DB2 catalog and save it in a library of your choice.

BROWSE

The BROWSE function is a utility that displays the contents of a DBRM for diagnostic purposes.

DECLARE TABLE exception

The DECLARE TABLE exception provides you with the option to disregard changes in DECLARE TABLE statements when comparing old and new DBRMs.

Coprocessor support

DB2 Bind Manager now supports compilers that use the coprocessor (integrated precompiler).

SC27-1450-00 (June 2001)

This section summarizes the significant improvements or enhancements for DB2 Bind Manager for z/OS, Version 2 Release 1:

- The Path Checker feature has been removed from this release of DB2 Bind Manager and is available as a separate product. DB2 Path Checker is Program Number 5655-E39.
- Timestamps in a DBRM are matched to those in a load module.
- A report of modules and CSECTs that contain static SQL can be obtained by a scan of a load module library.

Chapter 2. Configuration

Before you can use DB2 Bind Manager for the first time, you must configure the operating environment.

Before configuring DB2 Bind Manager, complete the installation instructions found in the *Program Directory for DB2 Bind Manager for z/OS*.

Topics:

- “Environment support”
- “Requirements”
- “Configuration summary” on page 14
- “Configuration steps” on page 14
- “Accessing DB2 Bind Manager from the DB2 Admin Launchpad” on page 21

Environment support

Before you configure DB2 Bind Manager, verify that your environment meets the following requirements.

DB2 Bind Manager and DBRM Checker support:

- DB2 Version 8 or Version 9
- REXX DB2 API
- Applications in VS COBOL, COBOL II, PL/I, Assembler, and C
- Diagnostics for DBRMs, DB2 catalog, and DB2 timestamp usage in packages
- Very large programs (greater than 10,000 statements)

DB2 Bind Manager runs in DB2I and batch environments.

Requirements

Before you configure DB2 Bind Manager, verify that your environment supports the following requirements.

DB2 Bind Manager requires:

- Access to the DBRM used by the most recent bind
- Any of the supported programming languages
- Any supported zSeries® processor running any supported release of z/OS

DBRM Checker requires:

- The names of the plans to be checked
- The high-level data set name qualifier of the DB2 subsystem
- The PDS library containing the DBRMs to be validated against the DB2 catalog tables

DBRM Cross-Reference requires:

- Access to the DBRM library that is to be checked
- Read access to the target DB2 subsystem

- Read access to the SYSIBM.SYSDBRM table in the referenced DB2 subsystem
- Read access to the load library that is to be verified

The DB2 load module cross-reference does not require APF authorization or *hooks* to DB2 or MVS™, and consists of the following load modules:

- BNDA006
- BNDA009 (works in the background to interface batch and ISPF programs to DB2 and must exist in the load library)

Load Module Static SQL requires:

- Read access to the load library

DB2 Bind Manager does not support PDSE libraries.

Configuration summary

After installing DB2 Bind Manager, complete the configuration steps that are summarized in the following table. For more information about each step, refer to the corresponding topics.

Table 1. Overview of configuration steps

Step	Description	Reference
1	Bind the DB2 Bind Manager plans.	“Step 1: Bind the DB2 Bind Manager plans.”
2	Change the module name.	“Step 2: Change the module name” on page 18.
3	Update JCL data strings for your installation.	“Step 3: Configure the JCL for your installation” on page 18.
4	Invoke DB2 Bind Manager from a foreground DB2 application (optional).	“Step 4: Configure DB2 Bind Manager for DB2I” on page 20.
5	Copy the BNDAVB module to an authorized library (optional).	“Step 5: Copy the BNDAVB module to an authorized library” on page 20.

Configuration steps

DB2 Bind Manager requires configuration steps needed for setup.

Step 1: Bind the DB2 Bind Manager plans

Edit and run the BNDBIND job for each DB2 subsystem where DB2 Bind Manager will be used. The BNDBIND member is in the SBNDSAMP library.

See the following BNDBIND sample job for values to be modified.

```

//BIND1 JOB ,ZZZ,CLASS=A,MSGCLASS=X,
//          NOTIFY=&SYSUID
//*
//* 5655-E43
//* (c) Copyright HLS Technologies, Inc. 2009 All Rights Reserved
//*
//*****
//*
//* This member binds the plans required for Bind Manager.
//*
//*
//*          IMPORTANT!
//*
//*          USE THIS JCL ONLY WHEN BINDING IN DB2 VERSION 8 OR LATER.

```

```

/**          FOR DB2 VERSION 7 OR EARLIER, USE BNDBIND7 INSTEAD.          *
**                                                                 **
/** Required changes:                                                                 **
/**                                                                 **
/** <SSID> - The DB2 subsystem ID                                                                 **
/** <AUTH> - The secondary authid to be used for creating tables *
/**          etc. (Optional parameter; defaults to user ID of the *
/**          individual submitting the job)                                                                 **
/** <OWNER> - The plan owner                                                                 **
/** <PLNQ> - The plan qualifier                                                                 **
/** <LLDS> - The name of the DB2 load library data set *
/**          installed in the target system (SDSNLOAD) *
/** <DBRL> - The name of the DB2 Bind Manager DBRM library *
/**          data set (SBNDDBRM) *
/** <SAMP> - The name of the DB2 Bind Manager sample library *
/**          data set (SBNDSAMP) *
/**                                                                 **
/** NOTE: This job must be run on each subsystem where DB2 Bind *
/** Manager will be used. *
/**                                                                 **
/** AAE5  AAFM  AAHH  AAHJ  AAHX *
**                                                                 **
/*******
/**
//CREATE EXEC  PGM=IKJEFT01
//STEPLIB   DD DISP=SHR,DSN=<LLDS>
//SYSPROC   DD DISP=SHR,DSN=<SAMP>
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN  DD *
           %BNDDFT <SSID> <AUTH>
/*
/**
//BIND EXEC  PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB   DD DISP=SHR,DSN=<LLDS>
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//DBRMLIB   DD DISP=SHR,DSN=<DBRL>
//SYSTSIN  DD *
DSN SYSTEM(<SSID>)

BIND PACKAGE(BNDV2R4)      -
  MEMBER(BNDAVB)           -
  OWNER(<OWNER>)           -
  QUALIFIER (<PLNQ>)      -
  ACTION(REPLACE)         -
  VALIDATE(BIND)          -
  ISOLATION(CS)           -

BIND PACKAGE(BNDBLD)      -
  MEMBER(BNDBLD)           -
  OWNER(<OWNER>)           -
  QUALIFIER (<PLNQ>)      -
  ACTION(REPLACE)         -
  VALIDATE(BIND)          -
  ISOLATION(CS)           -

BIND PACKAGE(BNDV2R4)      -
  MEMBER(BNDDBC)           -
  OWNER(<OWNER>)           -
  QUALIFIER (<PLNQ>)      -
  ACTION(REPLACE)         -
  VALIDATE(BIND)          -
  ISOLATION(CS)           -

BIND PACKAGE(BNDV2R4)      -

```

MEMBER(BNDA009)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
MEMBER(BNDA019)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
MEMBER(BNDA020)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
DYNAMICRULES(BIND)	-
ISOLATION(CS)	-
BIND PACKAGE(BNDV2R4)	-
MEMBER(BNDA021)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PACKAGE(BNDV2R4)	-
MEMBER(BNDBLDD)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
MEMBER(BNDA024)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
MEMBER(BNDA030)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
MEMBER(BNDA036)	-
OWNER(<OWNR>)	-
QUALIFIER (<PLNQ>)	-
ACTION(REPLACE)	-
VALIDATE(BIND)	-
ISOLATION(CS)	-
BIND PLAN(BNDV2R4)	-
PKLIST(BNDV2R4.*,*.BNDV2R4.*)	-

```

OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA009) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA019) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA020) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA024) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA030) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA036) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -
ISOLATION(CS) -
RETAIN

BIND PLAN(BNDA043) -
PKLIST(BNDV2R4.*,*.*.BNDV2R4.*) -
OWNER(<OWNR>) -
QUALIFIER (<PLNQ>) -
ACTION(REPLACE) -
VALIDATE(BIND) -

```

```

ISOLATION(CS)          -
RETAIN
END
/*
//

```

Figure 3. JCL sample job BNDBIND

To bind DB2 Bind Manager plans:

1. Edit the SBNDSAMP member BNDBIND according to the instructions.

Important: When DB2 objects are created, DB2 Bind Manager uses the value that is specified for <AUTH>. If a value is not specified for <AUTH>, the default value is the logon user ID. Therefore, if the owner who is specified in the CREATE step does not match the qualifier that is specified in the BIND step, the post-installation BNDBIND job will fail.

2. Run the BNDBIND member on the DB2 subsystem where DB2 Bind Manager is installed. This job creates table MINIMUM_DATA and binds the plan that is required to run DB2 Bind Manager.

Step 2: Change the module name

To use DB2 Bind Manager in the DB2 application compile procedure, change the module name in the precompile step from DSNHPC to BNDAVB, the DB2 Bind Manager module.

You also might need to modify condition testing on the bind job step to test for the DB2 Bind Manager return code.

Step 3: Configure the JCL for your installation

It might be necessary to configure the JCL for your installation.

The following example shows an example of the JCL for a typical DB2 compile job stream using DB2 Bind Manager. In the sample JCL, data strings that are in *italics* represent parameters that you must replace with values appropriate to your installation. For example, in the first job step specify in the INFILE DD statement DSN= the name and location of the existing DBRM for the program being compiled.

You also need to configure the remaining parameters, DB2 load library, DB2 Bind Manager load library, and temporary DBRM library, using the parameter name for your installation.

```

//Job Card
/*
//*****
//*                               Sample COBOL-II DB2 Compile          *
//*****
//* COPYDBRM steps tests for the existence of an old DBRM member    *
//* and if found copies it to a temporary DBRMLIB for processing    *
//* by DB2 Bind Manager. This step runs the TSO TMP in the         *
//* background. For new programs expect to get the message        *
//* IEC141I and a condition code of 12 because no DBRM exists.     *
//* This step is optional but recommended for production           *
//* environments.                                                  *
//*****
//*

```

```

//COPYDBRM EXEC PGM=IKJEFT01,
// PARM='REPRO IFILE(INFILE) OFILE(OUTFILE)'
//INFILE DD DISP=SHR,DSN=old.dbrmlib(dbrmname)
//OUTFILE DD DISP=SHR,DSN=temporary.dbrmlib(dbrmname)
//SYSTSPRT DD DUMMY
//*
//*****
//* Precompile the Source *
//* DSNHPC has been replaced with BNDAVB, *
//* DB2 Bind Manager STEPLIB added *
//*****
//*
//PC EXEC PGM=BNDAVB,PARM='HOST(COB2),APOST'
//STEPLIB DD DISP=SHR,DSN=db2.loadlib (if not in LINKLST)
//SYSLIB DD DISP=SHR,DSN=bndavb.loadlib
//DBRMLIB DD DISP=SHR,DSN=temporary.dbrmlib(dbrmname)
//SYSCIN DD DISP=(,PASS),DSN=&&COMPIN,UNIT=SYSALLDA,
// SPACE=(CYL,(5,2))
//SYSUT1 DD SPACE=(CYL,(4,2)),UNIT=SYSALLDA
//SYSUT2 DD SPACE=(CYL,(4,2)),UNIT=SYSALLDA
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//*
//*****
//* RC=0 The old.dbrmlib DBRM is still valid, no action needed. *
//* BIND not required *
//* RC=4 BIND required. Copy the DBRM from the temporary.dbrmlib *
//* to the old.dbrmlib. *
//*****
//* Execute COBOL-II compiler *
//*****
//*
//COB2 EXEC PGM=IGYCRCTL,COND=(4,LT),PARM='OBJECT,MAP,APOST'
//STEPLIB DD DISP=SHR,DSN=cobol2.loadlib (if not in LINKLST)
//SYSPRINT DD SYSOUT=*
//SYSIN DD DISP=(OLD,DELETE),DSN=&&COMPIN
//SYSLIN DD DISP=(MOD,PASS),UNIT=SYSALLDA,DSN=&&LOADSET,
// SPACE=(TRK,(3,3))
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT5 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT6 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//SYSUT7 DD UNIT=SYSALLDA,SPACE=(CYL,(3,2))
//*
//*****
//* Link Edit *
//*****
//*
//LKED EXEC PGM=IEWL,PARM='LIST,XREF,LET,MAP',COND=(5,LT,COB2)
//SYSLIN DD DISP=(OLD,DELETE),DSN=&&LOADSET
// DD ddname=SYSIN
//SYSLMOD DD DISP=SHR,DSN=loadlib(pgmname)
//SYSLIB DD DISP=SHR,DSN=syslib
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//*
//*****
//* Perform a bind if required *
//*****
//*
//BINDBRM EXEC PGM=IKJEFT01,DYNAMNBR=30,COND=((4,LT,LKED),(4,GT,PC))
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*

```

```

//DBRMLIB DD DISP=SHR,DSN=temporary.dbrmlib
//SYSTSIN DD *
  DSN SYSTEM(dsn)
  BIND plan/package control statements
END
/*
/**
/*******
/**      Copy new DBRM back to old DBRMLIB if bind was required      *
/*******
/**
//COPYBACK EXEC PGM=IEBGENER,COND=((4,LT,BINDDBRM),(4,GT,PC))
//SYSUT1 DD DISP=SHR,DSN=temporary.dbrmlib(dbrmname)
//SYSUT2 DD DISP=SHR,DSN=old.dbrmlib(dbrmname)
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY

```

Figure 4. Sample JCL for using DB2 Bind Manager

Step 4: Configure DB2 Bind Manager for DB2I

DB2 Bind Manager must be configured for DB2I.

To invoke DB2 Bind Manager from a foreground DB2 application:

1. Locate the CLIST named DSNH. (The default CLIST library for DB2I is named DSN ν rm.DSNCLIST, where ν rm is the DB2 version, release, and modification level.)
2. Within the CLIST, locate the statement that initializes the variable named PCLOAD. If the CLIST is the default version as shipped by IBM, the definition should look like this:

```
PCLOAD(''DSN $\nu$ rm.DSNLOAD(DSNHPC)'')
```

3. Make one of the following changes, depending on whether the DB2I load library is in the MVS link list (LINKLST).
 - a. If the DB2I load library is in LINKLST, modify the definition in the CLIST so that it looks like this:

```
PCLOAD(''your.loadlib(BNDAVB)'')
```

where *your.loadlib* is the name of the load library that contains DB2 Bind Manager.

- b. If the DB2I load library is not in LINKLST, move the BNDAVB load module to the DB2I load library, and modify the definition in the CLIST so that it looks like this:

```
PCLOAD(''DSN $\nu$ rm.DSNLOAD(BNDAVB)'')
```

Step 5: Copy the BNDAVB module to an authorized library

If you intend to use SMF generation, the BNDAVB must be run from an authorized library.

Generating SMF records requires APF authorization. Therefore, if you intend to use SMF generation, you must run the BNDAVB program from an APF-authorized library. Otherwise, this step is not necessary.

Accessing DB2 Bind Manager from the DB2 Admin Launchpad

If the DB2 Admin Launchpad is installed on your system, you can add an appropriate entry to the Launchpad table to enable access to DB2 Bind Manager from the DB2 Admin Launchpad.

To enable DB2 Bind Manager to be run from the DB2 Admin Launchpad:

1. Customize the CLIST in the sample member BNDADBI that is supplied as part of DB2 Bind Manager as described in the comments. The part of BNDADBI that you modify is located between the comments "BEGIN MODIFICATIONS HERE" and "END MODIFICATIONS HERE."
2. Run the CLIST to add DB2 Bind Manager to the Launchpad table. When the Launchpad is invoked, DB2 Bind Manager will be one of the choices displayed in the Launchpad tools list, and you will be able to execute it directly from there.

Chapter 3. DBRM and SQL analysis tools

The DBRM and SQL analysis tools include DBRM Checker, DBRM Load Module Cross-Reference, and Load Module Static SQL Report. The following topics provide more information about these tools:

Topics:

- “Using DBRM load module cross-reference”
- “Creating the static SQL load module report” on page 41
- “Catalog cleaning (program BNDA020)” on page 42
- “DBRM creation (program BNDA024)” on page 45
- “Catalog timestamp listing (program BNDA030)” on page 48
- “Catalog cleanup (program BNDA036)” on page 50
- “Batch DBRM interpreter (BNDA043)” on page 64
- “Catalog cleanup (program BNDA046)” on page 67

Using DBRM load module cross-reference

The DBRM load module cross-reference (hereafter called DBRM cross-reference) extends the logic of DBRM Checker to include load module comparisons.

You can use either ISPF or batch to run the program. ISPF allows for comparison of a single DBRM to a single load module. If the member is not listed on the panel, you can select from a list in DBRM or the load module library. After a DBRM and load module are selected, the DB2 subsystem is queried for its token for the DBRM. Results display in date/time format for the DBRM, the DB2 catalog and the load module.

Ensure that the following information is available to the DBRM cross-reference:

- SSID
- DBRMs from which tokens will be taken
- load modules to be checked

Related concepts:

“DBRM batch cross-reference (program BNDA006)” on page 36

Related tasks:

“Running ISPF mode”

Running ISPF mode

In ISPF mode you can compare a single DBRM to a single load module.

The DBRM cross-reference identifies the DB2 token in a DBRM library to the DB2 subsystem and to CSECTs within specific load modules.

Important: DB2 Bind Manager makes extensive use of ISPF Dialog Manager features, including scrollable fields. In DB2 Bind Manager, scrollable fields contain rulers to indicate which columns are currently visible. To view the remaining columns in a field, place the cursor in the field and issue the LEFT or RIGHT command to scroll in the indicated direction.

The default setting for the LEFT command is PF10, and the default setting for the RIGHT command is PF11.

To modify your environment to use DBRM cross-reference in ISPF mode, follow these steps:

1. Ensure that the LOADLIB (SBNDLOAD) contains BNDA005 and that it is accessible.
2. Ensure that the DB2 Bind Manager ISPF panels are in a library (SBNDPENU) that is accessible.
3. Ensure that CLIST BNDC002 is in a library (SBNDCLST) that is part of the SYSPROC concatenation in the TSO logon procedure.
4. Modify the CLIST BNDC002 code to conform to the local data set naming conventions that are described in the comments at the beginning of the CLIST, as shown in the following figure.

```

***** Top of Data *****
/* REXX */

/*
/* Licensed Materials - Property of IBM
/* 5655-E43
/* (c) Copyright IBM Corp. 2000, 2004 All Rights Reserved.
/* (c) Copyright HLS Technologies, Inc. 1991, 2004 All Rights
/* Reserved.
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
/*
/*
/* -----
/* IMPORTANT
/* -----
/*
/* Un-comment the following three statements
/* and modify them to reference the libraries
/* where the BND dialog components are
/* installed.
/*
/*
/* pnullib = 'BND.PNL' /* Name of BND panel library */
/* loadlib = 'BND.LOADLIB' /* Name of BND loadlib */
/* msglib = 'BND.MSG' /* Name of BND msg library */

address ISPEXEC
arg option .
if translate(option) = 'DEBUG' then do
SIGNAL ON NOVALUE
"CONTROL ERRORS CANCEL"
end
else
"CONTROL ERRORS RETURN"
trash = msg('OFF')
if pnullib = 'PNLLIB' | loadlib = 'LOADLIB' then do
call ISPMMSG 22
exit 12
end
if msglib = 'MSGLIB' then do
call ISPMMSG 22
exit 12
end
"LIBDEF ISPLLIB DATASET ID('pnullib')"
"LIBDEF ISPLLIB DATASET ID('msglib')"
"LIBDEF ISPLLIB DATASET ID('loadlib')"

```

Figure 5. CLIST BNDC002 code modification

5. Start the ISPF dialog by using standard TSO or ISPF syntax to run CLIST BNDC002. If you need help starting the dialog in your environment, contact your DBRM cross-reference installation specialist.

A call attach to the identified DB2 subsystem is performed, and the SYSIBM.SYSDBRM table is read and matched to the selected DBRMs. The selected load modules are matched against the DBRMs. The matching CSECT names are searched for the token in the same DBRM name.

When the CLIST BNDC002 is started, the DB2 Bind Manager main menu panel is displayed, as shown in the following figure:

```

-----
                                DB2 Bind Manager 2.3.0
COMMAND ===> _

Select the desired mode of operation

      0  SETTINGS  Defaults
      1  PLAN      Plan mode
      2  PKG       Package mode
      3  BROWSE    Browse DBRMs
      4  GENERATE  Rebuild DBRMs
      5  IXU       Index Utility
      X  EXIT      End dialog

Options:
/ Translate DB2 time stamps

```

Figure 6. DB2 Bind Manager main menu panel

Related tasks:

“Using DBRM load module cross-reference” on page 23

Related information:

 IBM Publications Center

Specifying default values

Use the Settings function to specify the DB2 Bind Manager default values.

To specify default values, use the following steps:

1. Select option **0 SETTINGS** on the main menu. The DB2 Bind Manager - SETTINGS panel is displayed, as shown in the following figure.

```

                                DB2 Bind Manager - SETTINGS
COMMAND ===> _                               Scroll ===> CSR

Default high level qualifier
for work files . . . . . DNS3

```

Figure 7. DB2 Bind Manager - SETTINGS panel

2. In the **Default high level qualifier for work files** field, specify the default high-level qualifier for work files. Depending on the function that is being performed, DB2 Bind Manager can use up to three work files (temporary data sets). DB2 Bind Manager requires the high-level qualifier to allocate the work files. The first time that you use the DB2 Bind Manager - SETTINGS panel, the current value is either the prefix from your TSO profile or your user ID.

Inspecting plans

Use the PLAN function to display the contents of a plan for diagnostic purposes.

To inspect plans, use the following steps:

1. Select option **1 PLAN** on the main menu. The DBRM Cross Check panel is displayed, as shown in the following figure.

```

----- DBRM Cross Check -----
COMMAND ==> _

DBRM      Data Set Name . . 'DEMO.DBRMLIB'
          Member . . . . . DSNTIAUL

LOAD      Data set name . . 'DEMO.LOADLIB'
MODULE    Member . . . . . DSNTIAD

DB2       Local . . . . . DB7G
Subsys    Remote . . . . .

```

Figure 8. DBRM Cross Check panel

2. Specify the following information, and press Enter.

DBRM

- In the **Data Set Name** field, specify the DBRM library in single quotation marks.
- In the **Member** field, specify the DBRM name.

LOAD MODULE

- In the **Data set name** field, specify the load module library in single quotation marks.
- In the **Member** field, specify the load module DBRM name.

DB2 Subsys

In the **Local** or **Remote** field, specify the DB2 subsystem ID.

If you omit any member names, a selection list is displayed, as shown in the following figure:

```

MEMBER LIST DEMO.DBRMLIB                      Row 00001 of 00002
COMMAND ==>                                Scroll==> PAGE
Name Prompt   Size  Created  Changed  ID
.DSNTIAD
.DSNTIAUL
**End**

```

Figure 9. DBRM module cross-reference selection list

The DBRM cross-reference results are displayed in the following figure. This panel displays timestamps for the converted DB2 tokens. If the token and load module timestamps do not match, the token information is not displayed.

```

                                DBRM Cross Check
COMMAND ==> _

-- Source --      ----- Time Stamp -----

DBRM              1999/09/08 21:48:16.654931
DB2 Catalog       2000/09/22 19:45:54.725860
Load Module       1999/09/08 21:48:16.654931

```

Figure 10. DBRM module cross-reference results

Inspecting packages

Use the PKG function to display the contents of a package for diagnostic purposes.

The information flow and display for packages and plans are the same except additional data fields that correspond to the collection and version names are included for the package process.

To inspect packages, use the following steps:

1. Select option **2 PKG** on the main menu. The Package Cross Check panel is displayed, as shown in the following figure.

```

----- Package Cross Check -----
COMMAND ===>

DBRM   Data Set Name . . . . 'DSN710.SDSNDBRM'
       Member . . . . . DSNREXX

PACKAGE Collection . . . . .
                               +-----+
                               | 1 18 |
                               +-----+

       Version . . . . .
                               +-----+-----+-----+
                               | 1 30 |
                               +-----+-----+-----+

LOAD   Data set name . . . . 'DSN710.SDSNLOAD'
MODULE Member . . . . . DSNREXX

DB2    Local . . . . . DB7G
Subsys Remote . . . . .

```

Figure 11. Package Cross Check panel

2. Specify the following information, and press Enter.

DBRM

- In the **Data Set Name** field, specify the name of the library that contains the DBRM.
- In the **Member** field, specify the DBRM name.

PACKAGE

- In the **Collection** field, specify the package collection.
- In the **Version** field, specify the version.

LOAD MODULE

- In the **Data set name** field, specify the load module library in single quotation marks.
- In the **Member** field, specify the load module DBRM name.

DB2 Subsys

In the **Local** or **Remote** field, specify the DB2 subsystem ID.

If the **Collection** and **Version** fields are left blank, all the collections and versions that include the subject DBRM are displayed. The following figure shows the panel with sample values specified.

```

Package Cross Check
DBRM Verification Results
Row 1 to 12 of 35
COMMAND ==> Scroll ==> CSR
BND345W Warning: multiple instances of DSNREXX found in DB7G
DBRM time stamp . . . . . : 2003/06/12 20:18:22.807058
Load module time stamp . . : 2003/06/12 20:18:22.807058

----- DB2 Subsystem (SYSIBM.SYSPACKAGE) -----
----- Timestamp ----- --- Collection --- ----- Version -----
1 18 1 26
-----+-----1-----+-----1-----2-----+
2000/09/22 18:09:11.782694 DSNREXCS Version (blank)
2001/05/30 23:25:47.740351 DSNREXCS Version UQ54704
2001/11/01 17:46:49.260458 DSNREXCS Version UQ59350
2002/09/27 23:16:17.709465 DSNREXCS Version UQ70197
2003/01/10 17:19:09.808300 DSNREXCS Version UQ72920
2003/02/06 18:03:19.711024 DSNREXCS Version UQ73910
2003/06/12 20:18:22.807058 DSNREXCS Version UQ77665
2000/09/22 18:09:11.782694 DSNREXRR Version (blank)
2001/05/30 23:25:47.740351 DSNREXRR Version UQ54704
2001/11/01 17:46:49.260458 DSNREXRR Version UQ59350
2002/09/27 23:16:17.709465 DSNREXRR Version UQ70197
2003/01/10 17:19:09.808300 DSNREXRR Version UQ72920

```

Figure 12. Sample DBRM verification results

3. Optional: To display a specific instance, specify the collection and version name on the Package Cross Check panel, as shown in Figure 11 on page 28. Sample DBRM verification results are shown in the following figure:

```

Package Cross Check
DBRM Verification Results
Row 1 to 1 of 1
COMMAND ==> Scroll ==> CSR
BND345W Warning: multiple instances of DSNREXX found in DB7G
DBRM time stamp . . . . . : 2003/06/12 20:18:22.807058
Load module time stamp . . : 2003/06/12 20:18:22.807058

----- DB2 Subsystem (SYSIBM.SYSPACKAGE) -----
----- Timestamp ----- --- Collection --- ----- Version -----
1 18 1 26
-----+-----1-----+-----1-----2-----+
2001/05/30 23:25:47.740351 DSNREXCS Version UQ54704
***** Bottom of data *****

```

Figure 13. Sample DBRM verification results for a specific instance panel

Analyzing a DBRM (BROWSE)

Use the BROWSE function to display the contents of a DBRM for diagnostic purposes.

To start the DBRM diagnostic display (DBRM BROWSE):

1. Select option 3 **BROWSE** on the main menu. The Browse DBRM panel is displayed, as shown in the following figure.

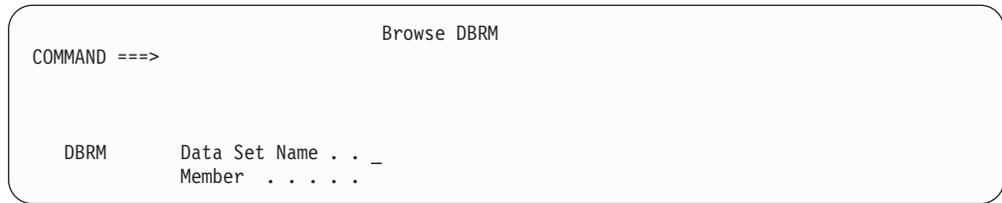


Figure 14. Browse DBRM panel

2. In the **Data Set Name** field, specify the name of the library that contains the DBRM in single quotation marks, in the **Member** field, specify the DBRM name (member name), and press Enter. The following figure shows the panel with sample values specified.



Figure 15. Sample DBRM selection panel

DBRM Checker reads the DBRM from the library and translates the contents into a readable format. The results are displayed in ISPF Browse as shown in the following figure:

```

-----
BROWSE      SYS04051.T230018.RA000.DNS3.R0103535      Line 00000000 Col 001 080
Command ==> _                                         Scroll ==> CSR
***** Top of Data *****

```

```

-----
DBRM Header
-----

```

```

Record ID (DBRMHID)          DBRM
Record Length (DBRMHLEN)    160
Precompile User ID (DBRMUSER) W98COMP
Program name (DBRMPROG)     DSNUGSQL
Precompile timestamp (DBRMTIMS) 2002.09.10.15.35.21.261969
                               ( 1706BA560327E45D hex )
Options (DBRMPOPT)         APOST,PERIOD,APOSTSQL,GRAPHIC(NO),
                               LANG=ASSEMBLER,TIME(LOCAL),DECIMAL(31),FOLD,
                               FLOAT(S390),SQL(DB2)
Character set ID (DBRMCCSID) 0000 (hex)
Max section number (DBRMMAXS) 0
Entry stmt format (DBRMTYPE) 1
Reserved (DBRMSCC)          0000 (hex)
Flags                        80 (hex)
Dependency marker (DBRMPDRM) N
DB2 Version (DBRMMRIC)      K (Version 7)

```

```

-----
Header Extension Record
-----

```

(No data)

```

-----
SQL statements
-----

```

```

Section 0
Statement nbr 108
SQL statement:
  DECLARE S1 STATEMENT
Host variables referenced by this statement:
  (None)

```

```

Section 1
Statement nbr 109
SQL statement:
  DECLARE C1 CURSOR WITH HOLD FOR S1
Host variables referenced by this statement:
  (None)

```

```

Section 1
Statement nbr 158
SQL statement:
  PREPARE S1 INTO : H FROM : H
Host variables referenced by this statement:
  TYPE=Output      VARIABLE=SQLDA
  TYPE=Undefined   VARIABLE=SQLSTMT

```

Figure 16. DBRM output

Re-creating a DBRM (GENERATE)

With DBRM Rebuilder (GENERATE), you can re-create a plan- or package-based DBRM that is missing from the DB2 catalog and save it in a library that you specify.

The DBRM Rebuilder can re-create a DBRM based on either a plan or a package.

Important: The following steps show how to re-create a DBRM based on a plan, but most of the steps for re-creating a DBRM based on a package are the same.

To re-create a DBRM, use the following steps:

1. Select option **4 GENERATE** on the main menu. The DBRM Rebuilder panel is displayed, as shown in the following figure.

```
                                DBRM Rebuilder
COMMAND ==>> _
Select one of the following options:
      1 PLAN   Plan mode
      2 PKG   Package mode
      X EXIT   End dialog
Subsystem ID . . . . . DSN1
```

Figure 17. The DBRM Rebuilder panel

2. Select option **1 PLAN** , specify the DB2 subsystem ID in the **Subsystem ID** field, and press Enter. The DBRM Rebuilder Plan/DBRM selection panel is displayed, as shown in the following figure.

```
                                DBRM Rebuilder
COMMAND ==>> _
Specify plan and DBRM names
Plan name . . . . .
DBRM name . . . . .
```

Figure 18. Plan/DBRM selection panel

3. Specify the following information, and press Enter.

Plan name

The plan from which to extract the DBRM. This plan is the same plan into which the DBRM was bound in the specified DB2 subsystem.

DBRM name

The DBRM that you want to re-create.

The following figure shows the panel with sample values specified. In this example, the DBRM *sampdbrm* is re-created from a plan with the same name.

```

                                DBRM Rebuilder
COMMAND ==>>
Specify plan and DBRM names
    Plan name . . . . . sampdbrm
    DBRM name . . . . . sampdbrm _

```

Figure 19. Sample Plan/DBRM selection panel

The DBRM library panel is displayed.

4. In the **DBRM library** field, specify the output library (DBRMLIB) in single quotation marks, and press ENTER. The output library is the DBRM library where the finished DBRM is to be stored. The example in the following figure shows that DNS1.DBRMLIB is specified as the output library.

```

                                DBRM Rebuilder
COMMAND ==>>
Specify the output DBRM library
    DBRM library . . . . . 'DNS1.DBRMLIB' _

```

Figure 20. Specifying the output library

The DBRM Rebuilder extracts the DBRM information from DB2, formats it into a DBRM, and stores it in the designated output library. After the DBRM is stored in the output library, the Figure 18 on page 32 is displayed with the message *DBRM created* displayed, as shown in the following figure.

```

                                DBRM Rebuilder                DBRM created
COMMAND ==>> _
Specify plan and DBRM names
    Plan name . . . . . SAMPDBRM
    DBRM name . . . . . SAMPDBRM

```

Figure 21. DBRM created

5. Optional: You can use the DBRM Browse function to inspect and verify the re-created DBRM. The following figure shows a sample review of the re-created DBRM.

```

-----
BROWSE      SYS04051.T222657.RA000.DNS3.R0103533      Line 00000000 Col 001 080
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
-----
DBRM Header
-----

Record ID (DBRMHID)          DBRM
Record Length (DBRMHLEN)    160
Precompile User ID (DBRMUSER) $REBUILT
Program name (DBRMPROG)     SAMPDBRM
Precompile timestamp (DBRMTIMS) 2004.02.13.21.55.24.442069
                               ( 1758A66B02FA7578 hex )
Options (DBRMPOPT)         APOST,PERIOD,APOSTSQL,GRAPHIC(NO),
                               LANG=ASSEMBLER,TIME(LOCAL),DECIMAL(31),FOLD,
                               FLOAT(S390),SQL(DB2)

Character set ID (DBRMCCSID) 0000 (hex)
Max section number (DBRMMAXS) 0
Entry stmt format (DBRMSTYPE) 1
Reserved (DBRMSCC)          0000 (hex)
Flags                        80 (hex)
Dependency marker (DBRMPDRM) N
DB2 Version (DBRMMRIC)      K (Version 7)

-----
Header Extension Record
-----

```

Figure 22. Inspecting the re-created DBRM

Determining available indexes for tables (Index utility)

Use the online Index utility function to quickly determine the indexes available for a table.

To start the Index utility, use the following steps:

1. Select option **5 IXU** from the main menu. The DB2 Bind Manager Index Utility panel is displayed, as shown in the following figure. The following figure shows the panel with sample values specified.

```

                                DB2 Bind Manager Index Utility
COMMAND ==>

Specify the table whose indexes are to be displayed:

Subsystem ID . . . . . DB7G

Table name . . . . . SYSIBM.SYSCOLUMNS
                -----1-----2-----
                1      27

```

Figure 23. The DB2 Bind Manager Index Utility panel

2. In the **Subsystem ID** field, specify the DB2 subsystem ID, in the **Table name** field, specify the name of the table for the indexes that are to be displayed, and press Enter. DB2 Bind Manager queries the specified subsystem catalog to find the indexes that are associated with the named table. The indexes are shown on the Available Indexes panel, shown in the following figure.

```

Available Indexes
Row 1 to 2 of 2
COMMAND ==> Scroll ==> PAGE

Indexes defined for SYSIBM.SYSCOLUMNS:

----- Index Name -----
. SYSIBM.DSNDXC01
. SYSIBM.DSNDXC02
***** Bottom of data *****
----- Index Key Columns -----
TBCREATOR (A), TBNAME (A), NAME (A)
TYPESCHEMA (A), TYPENAME (A)

```

Figure 24. Available Indexes panel

The Available Indexes panel lists each index that is defined for the subject table. The name of the index is shown with the names of the key columns in order from major to minor. Each key column is followed by the letter A (ascending) or D (descending), which indicates whether the collating sequence for the column is ascending or descending. Only 40 display columns are available for the listing of key columns. If the number of key columns is not adequate to accommodate all of the information, the information is truncated.

- Optional: To view all of the information, select the index by specifying an s beside the index name, and press Enter. The panel in the following figure shows an example of a selected index name. All the key columns for the specified index are displayed in a scrollable list.

```

Available Indexes
Row 1 to 2 of 2
COMMAND ==> Scroll ==> PAGE

Indexes defined for SYSIBM.SYSCOLUMNS:

----- Index Name -----
s SYSIBM.DSNDXC01
. SYSIBM.DSNDXC02
***** Bottom of data *****
----- Index Key Columns -----
TBCREATOR (A), TBNAME (A), NAME (A)
TYPESCHEMA (A), TYPENAME (A)

```

Figure 25. Available indexes with a selected index

The following figure shows an example of the Index key columns panel.

```

Available Indexes
Row 1 to 2 of 2
COMMAND ==> ==> PAGE

Indexes defin
COMMAND ==>
Row 1 to 3 of 3
Scroll ==> PAGE

-----
s SYSIBM.D
. SYSIBM.D
*****
Index key columns
Seq Order Column
1 A TBCREATOR
2 A TBNAME
3 A NAME
***** Bottom of data *****
-----
ME (A)
*****

```

Figure 26. Index key columns panel

DBRM batch cross-reference (program BNDA006)

BNDA006 is the DB2 Bind Manager batch cross-reference program. It compares timestamps of the DBRMs that are in a DB2 library to the consistency tokens in a DB2 catalog and in one or more load libraries, and it attempts to find matches.

The BNDA006 program does the same comparison as the ISPF mode of load module cross-reference. Additionally, with the batch cross-reference program you can make many comparisons in a single run instead of doing each comparison one by one.

The batch process is controlled by JCL and several commands, which support wildcard characters. You can use wildcard characters to analyze an entire library or just a subset of the members.

Related tasks:

“Using DBRM load module cross-reference” on page 23

BNDA006 job control statements (JCL)

Various job control statements are associated with the BNDA006 program, including EXEC, STEPLIB DD, and SYSPRINT DD.

The following table shows the job control statements for program BNDA006.

Table 2. Job control statements for BNDA006

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. The EXEC statement is issued in the following format: <pre>//stepname EXEC PGM=BNDA006</pre> PGM=BNDA006 specifies that you want to run the DBRM batch cross-reference program.
STEPLIB DD	The STEPLIB DD statement defines an existing load library that contains the DB2 Bind Manager programs. This DD statement is required unless BNDA006 was installed in the system link list. It must reference the load library that contains the DB2 Bind Manager programs.
SYSPRINT DD	The SYSPRINT DD statement is required. It defines a sequential data set that is used to list control statements and messages. You can assign SYSPRINT to SYSOUT or to any sequential data set.
DBRM nn DD	The DBRM nn DD statement defines an existing partitioned data set that contains the DBRMs to be checked. In this statement, nn is a number 00 - 99. You can specify any number of DBRM nn DD statements, but each DD name must be unique. The numbers (nn) do not need to be sequential.
LOAD nn DD	The LOAD nn DD statement defines an existing load library that contains the load modules to be checked. In this statement, nn is a number 00 - 99. You can specify any number of LOAD nn DD statements, but each DD name must be unique. The numbers (nn) do not need to be sequential.
SYSIN DD	The SYSIN DD statement is required. It defines the control data set. It must precede a set of valid BNDA024 commands.

Sample JCL for BNDA006

Sample JCL for BNDA006 is provided in SAMPLIB member BNDJ006. The sample JCL is shown in the following figure.

```
//BNDJ006 JOB (ACCT),' ',CLASS=?,MSGCLASS=?
//*
//* 5655-E43
//* (c) Copyright HLS Technologies, Inc. 2001, 2009 All Rights Reserved
//*
//*****
//***** CHANGE LOG *****
//*****
//*
//* AAHJ 20070124 DNS
//* AAJG 20071119 DNS
//* V2R4 20090607 DNS
//* AAK9 20090722 DNS
//*
//*****
//*
//RUN EXEC PGM=BNDA006
//STEPLIB DD DISP=SHR,
// DSN= *** PDS CONTAINING BNDA006 PROGRAM ***
// DD DISP=SHR,
// DSN= *** PDS CONTAINING DB2 LOAD LIBRARY ***
//DBRM00 DD DISP=SHR,DSN= *** DBRMLIB TO BE PROCESSED ***
//DBRM01 DD DISP=SHR,DSN= *** DBRMLIB TO BE PROCESSED ***
//DBRM02 DD DISP=SHR,DSN= *** DBRMLIB TO BE PROCESSED ***
//*
//LOAD00 DD DISP=SHR,DSN= *** LOADLIB TO BE PROCESSED ***
//LOAD01 DD DISP=SHR,DSN= *** LOADLIB TO BE PROCESSED ***
//LOAD02 DD DISP=SHR,DSN= *** LOADLIB TO BE PROCESSED ***
//*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
* MEMLIST MEMLIST=YES to list included and OPTIONAL
* excluded modules (default)
* MEMLIST=NO to suppress listing of
* included and excluded modules
* SSID= SUBSYSTEM ID REQUIRED
* DBRM=DSN* DBRM NAME(S) OPTIONAL
* LMOD=DSN8* LOAD MODULE NAME(S) OPTIONAL
*
* NOTE: DBRM AND LOAD MODULE NAMES
* MAY INCLUDE WILDCARDS
/*
//
```

Figure 27. Sample JCL for BNDA006

BNDA006 commands

The BNDA006 program recognizes specific command statements that conform to defined syntax rules.

BNDA006 commands are read from SYSIN. BNDA006 commands must conform to the following syntax rules:

- Comment lines are indicated by an asterisk (*) in column one and are ignored.
- Comment lines can be interspersed with command lines in any order.
- You can enter an unlimited number of comment lines.
- Commands must be wholly contained between columns 1 - 72. Within this range, commands are free-form and can start in any column.

- Each command must be contained on a single line.

Wildcard characters

You can use wildcard characters with some commands to specify a range of one or more objects to be processed. The individual command descriptions indicate whether you can use wildcard characters.

Where wildcard characters are supported, the following conventions are supported:

- A percent sign (%) represents any single character.
- An asterisk (*) represents any number of characters.

Commands

The BNDA006 program recognizes the following commands:

SSID

The SSID command specifies the DB2 subsystem catalog against which DBRMs are to be compared. This command is optional and does not have a default. Therefore, if this command is not present, the DB2 catalog is not verified, and only the DBRM/load module is correlated.

The SSID command is issued in the following format:

```
SSID=name
```

where *name* is the name of the local DB2 subsystem. The *name* variable is either the subsystem whose catalog you want to check (that is, the subsystem in which the plans and or packages that you are checking were bound) or a subsystem that has a valid DDF connection to a remote subsystem whose catalog is the real target.

Specify a 1- to 4-character subsystem ID.

Restriction:

- Wildcard characters are not supported for the SSID command.
- The SSID command can be issued only once per run.

REMOTESYS

The REMOTESYS command specifies a catalog on a remote subsystem against which DBRMs are to be compared.

The REMOTESYS command is issued in the following format:

```
REMOTESYS=name
```

where *name* is the name of the remote DB2 subsystem whose catalog you want to check (that is, the subsystem in which the plans and packages you are checking were bound).

Specify a 1- to 16-character subsystem ID. This command is optional and does not have a default. Therefore, if this command is not present, a remote subsystem is not accessed.

The REMOTESYS command requires you to process a valid SSID command. The SSID command specifies the local system whose DDF connection is used to access the remote system that is specified by REMOTESYS. To access a remote subsystem, you must specify both the

SSID command and the REMOTESYS command. If the local system is the target, then specify only the SSID command.

Restriction:

- Wildcard characters are not supported for the REMOTESYS command.
- The REMOTESYS command can be issued only once per run.

MEMLIST

The MEMLIST command turns on or turns off the listing of DBRMs and load modules to be included and excluded from consideration in the cross-reference analysis.

The MEMLIST command is issued in the following format:

```
MEMLIST=YES  
(or)  
MEMLIST=NO
```

MEMLIST=YES specifies that the included and excluded DBRMs and load modules are to be included in the report. **MEMLIST=NO** specifies that the information is to be omitted. The default value for the MEMLIST command is YES.

DBRM

The DBRM command specifies the DBRMs to be checked. The DBRMs are retrieved from the libraries that are specified by the *DBRM_{mn}* DD statements.

The DBRM command is issued in the following format:

```
DBRM=dbrmname
```

where *dbrmname* is either the name of a specific DBRM or a pattern that includes one or more wildcard characters. Only DBRMs that match the specified name or pattern are checked. Any DBRMs that are present in *DBRM_{mn}* libraries but do not match the pattern are ignored.

Specify a 1- to 8-character name or pattern. If this command is not present, the default is to check all DBRMS (which has the same effect as specifying *DBRM=**).

You can issue the DBRM command as many times as necessary. For example, if the following commands are specified:

```
DBRM=HAROLD  
DBRM=SYS*
```

BNDA006 analyzes the DBRM named HAROLD and all DBRMs that are found with names that begin with SYS.

LMOD

The LMOD command specifies the load modules to be checked. The load modules are retrieved from the libraries that are specified by the *LOAD_{nn}* DD statements.

The LMOD command is issued in the following format:

```
LMOD=lmname
```

where *lmname* is either the name of a specific load module or a pattern that includes one or more wildcard characters. Only load modules that match

the specified name or pattern are checked. Any load modules that are present in LOADLIB but do not match the pattern are ignored.

Specify a 1- to 8-character name or pattern. If this command is not present, the default is to check all load modules (which has the same effect as specifying LMOD=*).

You can issue LMOD command as many times as necessary. For example, if the following commands are specified:

```
LMOD=MELVIN
LMOD=DSN*
```

BNDA006 analyzes the load module named MELVIN and all load modules that are found with names that begin with DSN.

Related reference:

“BNDA006 job control statements (JCL)” on page 36

Interpreting BNDA006 output

The BNDA006 program produces a report based on the commands that you specify in the program JCL.

The following figure shows an example of a batch cross reference report.

08/27/09 23:03:26		DBRM/Load Module Cross Reference						PAGE 0001
DBRM	DD	DBRM Time Stamp	--Referenced by-- LMOD CSECT	----- DB2 Subsystem ----- DBRM Present Timestamps Agree	----- DB2 Subsystem ----- DBRM Present Timestamps Agree	----- DB2 Subsystem ----- DBRM Present Timestamps Agree	Messages	
DSN0SCHL	03	2000/06/30 16:39:34	-- -- -- --				Not referenced by any load module	
DSN0SCJ	03	2000/06/09 19:09:02	DSNZCMD DSNAA				In LOAD00	
DSN0SM68	03	1991/12/19 07:01:51	-- -- -- --				Not referenced by any load module	
DSN0SPCL	03	2006/11/15 01:37:48	-- -- -- --				Not referenced by any load module	
DSN0SPC8	03	2007/09/25 18:24:43	-- -- -- --				Not referenced by any load module	
DSN0SPM	03	2003/11/08 23:02:40	-- -- -- --				Not referenced by any load module	
DSN0SPMN	03	2003/07/08 23:08:51	-- -- -- --				Not referenced by any load module	
DSN0STAT	03	2006/11/15 01:38:03	-- -- -- --				Not referenced by any load module	
DSN0STAB	03	2007/09/25 18:24:47	-- -- -- --				Not referenced by any load module	
DSNZSCJ	03	2000/06/09 19:09:02	-- -- -- --				Not referenced by any load module	
BNDA019	04	2009/06/19 00:39:41	BNDA019 BNDA019	Yes	Yes		In LOAD06	
C-BNDV2R4								
V-2009-06-19-00.39.41.657493								
BNDA043	04		-- -- -- --				*** Empty DBRM (no data) ***	
BNDA046	04	2009/08/16 17:54:01	BNDA046 BNDA0461				In LOAD06	
BNDBLD	04	2009/07/20 19:28:56	BNDBLD BNDBLD	Yes	Yes		In LOAD06	
C-BNDV2R4								
V-2009-07-20-19.28.56.957930								
BNDBLDD	04	2009/06/19 17:04:52	BNDBLDD BNDBLDD	Yes	Yes		In LOAD06	
C-BNDV2R4								
V-2009-06-19-17.04.52.723960								
BNDDBC	04	2009/06/19 17:05:00	BNDDBC BNDDBC	Yes	Yes		In LOAD06	
C-BNDV2R4								
V-2009-06-19-17.05.00.084206								
BNDX001	04	2009/05/22 18:07:38	BNDX001 BNDX0011	Yes			In LOAD06	
BNDAVB	05	2009/04/11 20:55:35	BNDAVB BNDAVB	Yes			In LOAD07	
BNDVAVBX	05	2004/05/11 22:21:32	-- -- -- --				Not referenced by any load module	
BNDA009	05	2008/07/02 22:56:57	BNDA009 BNDA009	Yes			In LOAD07	
BNDA019	05	2005/12/23 18:20:06	BNDA019 BNDA019	Yes	Yes		In LOAD07	
C-BND19C								
BNDA020	05	2008/05/01 17:58:54	-- -- -- --	Yes			Not referenced by any load module	
BNDA021	05	2007/07/27 23:22:21	-- -- -- --	Yes			Not referenced by any load module	
BNDA024	05	2007/11/08 01:31:06	BNDA024 BNDA0241	Yes	Yes		In LOAD07	
BNDA030	05	2006/06/22 03:11:40	BNDA030 BNDA0301	Yes			In LOAD07	
BNDA036	05	2009/07/25 01:20:36	BNDA036 BNDA0361	Yes			In LOAD07	
BNDA043	05		-- -- -- --				*** Empty DBRM (no data) ***	
BNDA046	05	2007/11/26 20:11:25	BNDA046 BNDA0461				In LOAD07	
BNDBLD	05	2008/01/29 17:53:42	BNDBLD BNDBLD	Yes			In LOAD07	
BNDBLDD	05	2007/07/27 23:28:55	-- -- -- --	Yes			Not referenced by any load module	
BNDDBC	05	2009/03/16 19:10:43	BNDA043 BNDDBC	Yes	Yes		In LOAD07	
C-BND43C								
BNDIAG	05	2006/01/06 03:14:43	BNDIAG BNDVAVB				In LOAD07	
BNDX001	05	2008/06/13 05:32:55	BNDX001 BNDX0011	Yes	Yes		In LOAD07	
HIR001	05		-- -- -- --				*** Empty DBRM (no data) ***	
HLSA009	05	2007/12/31 22:32:19	HLSA009 HLSA009	Yes	Yes		In LOAD07	
HLSA019	05	2007/12/31 22:32:32	HLSA019 HLSA019	Yes	Yes		In LOAD07	
C-HLS19C								
HLSA030	05	2009/08/25 23:11:28	HLSA030 HLSA0301	Yes			In LOAD07	
HLSA036	05	2009/08/25 23:11:43	HLSA036 HLSA0361	Yes			In LOAD07	
HLSA043	05		-- -- -- --				*** Empty DBRM (no data) ***	
HLSA046	05	2009/08/25 23:12:14	HLSA046 HLSA0461	Yes			In LOAD07	
HLSA901	05	2004/12/09 01:46:26	-- -- -- --				Not referenced by any load module	

Figure 28. BNDA006 sample cross-reference report

The output provides the following information:

| **DBRM**

| Indicates the name of the DBRM.

| **DD**

| Indicates the library from which the DBRM was taken. For example, 00
| indicates DD statement DBRM00, and 01 indicates DD statement DBRM01.

| **DBRM time stamp**

| Indicates the DB2 consistency token from the DBRM.

| **LMOD**

| Indicates the load module or modules that reference the DBRM consistency
| token.

| **CSECT**

| Indicates the CSECT within the load module where the matching
| consistency token is found.

| **DBRM Present**

| Indicates whether the DBRM is present in the subsystem.

| **Timestamps Agree**

| Indicates whether the DBRM and the subsystem timestamps match.

|
| When a DBRM matches one or more package entries in the catalog, the collection
| name is displayed on the next line. If the version identifier is not blank, it is shown
| also.

| The sample output indicates the following information:

- |
- DSNZSCJ does not match any catalog entry.
 - BNDX001 was found in the catalog but does not have a collection or version ID.
 - BNDBLD was found in the catalog in collection BNDV2R4, version
| 2009-06-19-17.05.00.084206.

Creating the static SQL load module report

The static SQL Load module (BNDA007) produces a report that shows which load module and CSECTs within the load module call static SQL. This report is useful when you attempt to match DBRMs with load modules.

The static SQL load module reads all members in a load library and examines each CSECT within each load module. If a CSECT uses static SQL, the report shows the load module name and the CSECT name. It also places a YES in the SQL column. If SQL use is not found, a NO is placed in the SQL column.

The sample JCL that is shown in the following figure produces the report that is shown in Figure 30 on page 42.

```

//BND A007 JOB (ACCT),' ',CLASS=?,MSGCLASS=?
//*

//*****
//***** CHANGE LOG *****
//*****
//*
//* ---- 20010416 DNS
//*
//*****
//*
//RUN EXEC PGM=BND A007
//STEPLIB DD DISP=SHR,
// DSN= *** PDS CONTAINING BND A007 PROGRAM ***
//LOADLIB DD DISP=SHR,
// DSN= *** LOADLIB TO BE PROCESSED ***
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//

```

Figure 29. Sample JCL for static SQL load module report

```

                                Load Modules Containing Static SQL
                                PAGE
                                0001

0Load Library: P390.BND210.LOADLIB

0--LMOD--  CSECT      SQL?

BND AVB                No
BND A001                No
BND A002                No
BND A002A               No
BND A002B               No
BND A003                No
BND A005                No
BND A006                No
BND A007                No
BND A009      BND A009  Yes
BND CKSUB              No

```

Figure 30. Static SQL load modules

Catalog cleaning (program BNDA020)

Program BNDA020, one of the DB2 Bind Manager catalog clean-up programs, correlates the package entries in the DBRM catalog to one or more load libraries to determine which packages are still referenced by load modules.

BNDA020 works in DB2 V7 and earlier versions.

For each DBRM that is not referenced by a load module, the FREE command is generated and written to a sequential data set. You can issue the commands at a later time to remove the orphan package entries from the catalog.

Due to the inherently high resource consumption that is involved in this type of processing, the catalog clean-up function runs as a batch job. The job is controlled by JCL.

BNDA020 job control statements (JCL)

Various job control statements are associated with the BNDA020 catalog clean up program.

Member BNDJ020 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

The following table shows the job control statements for BNDA020.

Table 3. Job control statements for BNDA020

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. It is issued in the following format: <code>//stepname EXEC PGM=BNDA020</code> where PGM=BNDA020 specifies that you want to run the catalog clean up program.
STEPLIB DD	The STEPLIB DD statement must reference the load library that contains the DB2 Bind Manager programs. This DD statement is required unless the DBRM Checker programs were installed in the system link list.
SYSOUT DD	The SYSOUT DD statement defines a sequential data set that is used for error and diagnostic messages. Normally this data set is empty. If a failure or malfunction occurs, include the contents of SYSOUT as part of the problem description.
SYSPRINT DD	The SYSPRINT DD statement defines a sequential data set that is used for messages and output reports.
DBRMOUT DD	The DBRMOUT DD statement defines a sequential data set where backup DBRMs, if any, are written. If backup DBRMs are to be saved, this statement must be present. DISP=MOD is required because the data set is opened and closed multiple times. The data set must have the attributes DSORG=PS, RECFM=FB, and LRECL=80.
BNDFCMD DD	The BNDFCMD DD statement defines a sequential data set where the generated FREE commands are written. The data set must have the attributes DSORG=PS, RECFM=FB, and LRECL=200.
BNDIN DD	The BNDIN DD statement defines the control data set. It must be followed by a set of valid commands. This DD statement is required.
SYS <i>nnn</i> DD	The SYS <i>nnn</i> DD statement defines an existing load library that contains load modules to be checked. The ddnames must be of the format SYS <i>nnn</i> , where <i>nnn</i> is three numeric digits. Any library that is identified by such a DD statement is searched. You can specify the SYS <i>nnn</i> statements in any order, and the numbers (<i>nnn</i>) do not need to be consecutive or contiguous. Duplicates are not allowed.

BNDA020 commands

The BNDA020 program recognizes specific command statements that conform to defined syntax rules.

BNDA020 command statements are read from SYSIN. BNDA020 commands conform to the following syntax rules:

- You must enter commands in the format `COMMAND(operand)`.
- Embedded blanks are not allowed.
- Each command must be entered on a separate line in columns 1 - 72.

The BNDA020 program recognizes the following commands:

COLLECTION

The following example shows the COLLECTION command format:

```
COLLECTION(name)
```

Specify the collection name of the packages to be checked. For example, if you enter the command COLLECTION(HRDB700), the catalog search is limited only to packages included in collection HRDB700.

This command is optional and can be specified with either the DSN or OWNER commands, or both.

COMMIT (*nnnn*)

Use the COMMIT command to specify the number of SQL calls that are permitted between COMMITs. A valid value is a 4-digit number 0002 - 9999. If an invalid entry is detected, the default value (0100) is assumed.

Some installations run monitoring software that automatically cancels tasks that issue too many SQL calls without an intervening COMMIT. The purpose of this command is to protect BNDA020 from being canceled in this way. If your system is using such software, issue this command with a value that is at least one less than the cancellation threshold of the software.

In some environments, frequent COMMITs have a noticeable impact on performance. Setting the COMMIT command value too low might result in degraded performance on a busy system. You must set the value as high as possible without risking cancellation by the monitoring software.

DSN The following example shows the DSN command format:

```
DSN(ssid)
```

Specify the DB2 subsystem (*ssid*) of the catalog to be searched. For example, if you enter the command DSN(DB7G), you are specifying subsystem DB7G as the target.

This command is required.

OWNER

The following example shows the OWNER command format:

```
OWNER(name)
```

Specify the owner ID of the catalog entries to be checked. For example, if you enter the command OWNER(TS58), the catalog search is limited only to packages owned by the user TS58.

This command is optional and can be specified with either the DSN or COLLECTION commands, or both.

Each command can be specified only once per run.

A FREE command is generated for any package in this subsystem that fits the specified criteria (OWNER and COLLECTION) and that is not matched by a load module.

BNDA020 excludes any modules with names that begin with DSN, DGO, DSQ, SYS, or SQL. That is, it does not generate FREE commands for those modules. BNDA020 creates FREE commands for all other packages for which it cannot find matching load modules. Except for these exclusions, FREE commands are generated for all non-matching packages, including those packages that were

created by remote binds or BIND COPY. However, because of the way remote binds and BIND COPY work, the packages they create do not contain sufficient information to allow the creation of backup DBRMs. If a DBRM backup is requested for such a package, an error message is issued and the program continues.

DBRM creation (program BNDA024)

Use the BNDA024 batch program to create DBRMs from the catalog.

The BND024 program is similar to the GENERATE function. By using BNDA024, you can generate multiple DBRMs in a single run instead of generating them one at a time, as you are restricted to online.

Related tasks:

“Re-creating a DBRM (GENERATE)” on page 32

BNDA024 job control statements (JCL)

Various job control statements are associated with program BNDA024.

Member BNDJ024 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

The following table shows the job control statements for BNDA024.

Table 4. Job control statements for BNDA024

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. It is issued in the following format: //stepname EXEC PGM=BNDA024 where PGM=BNDA024 specifies that you want to run the BNDA024 DBRM generator program.
STEPLIB DD	The DD statement must reference the load library that contains the DB2 Bind Manager programs. This DD statement is required unless BNDA024 was installed in the system link list.
SYSOUT DD	The SYSOUT DD statement defines a sequential data set that is used for listing control statements and messages. BNDA024 writes a log of control statements and status messages to the SYSOUT DD statement. This DD statement is required. You can assign SYSOUT to SYSOUT or to any sequential data set.
SYSPRINT DD	The SYSPRINT DD statement defines a sequential data set that is used for output reports. Reports that are generated by the LIST command are written to the SYSPRINT DD statement. This DD statement is required. You can assign SYSPRINT to SYSOUT or to any sequential data set.
DBRMLIB DD	The DBRMLIB DD statement defines an existing partitioned data set where the generated DBRMs are to be stored. This statement is required.
SYSUT1 DD	The SYSUT1 DD statement defines a work file. This statement is required.
SYSIN DD	The SYSIN DD statement defines the control data set. This statement must precede a set of valid BNDA024 commands. This statement is required.

BNDA024 commands

The BNDA024 program recognizes specific command statements that must adhere to defined syntax rules.

BNDA024 command statements are read from SYSIN. BNDA024 commands must adhere to the following syntax rules:

- Comment lines are indicated by an asterisk (*) in column one and are ignored.
- Comment lines are interspersed with command lines in any order. There is no limit to the number of comment lines that are allowed.
- Commands must be wholly contained between columns 1 and 72. Within this range, commands are free form and can start in any column.
- Each command must be contained on a single line. Continuing the command on a second line or multiple-line commands are not allowed.

Wild cards

Some commands permit the use of wild card characters to specify a range of one or more objects to be processed. The individual command descriptions indicate when you can use wild card characters. Where wild cards are allowed, the following conventions are supported:

- A percent sign (%) represents any single character.
- An asterisk (*) represents any number of characters.

Commands

The BNDA024 program recognizes the following commands:

DBRM

The DBRM command directs the program to re-create a single DBRM from a plan by using information from the DB2 catalog that is specified by the SSID statement. This statement is optional.

The DBRM command format is:

```
DBRM plan.dbrm
```

Specify the name of the plan and the specific DBRM within the plan. The regenerated DBRM is stored in the data set that is pointed to by ddname DBRMLIB under member name DBRM.

For example, the statement, DBRM PAYROLL.WITHHOLD causes DBRM WITHHOLD from plan PAYROLL to be re-created as member WITHHOLD in the library that is referenced by DBRMLIB.

DEFAULTS

You can provide default values for the DB2 version and host language by using the DEFAULTS command. This statement is optional. Some DBRMs, particularly those generated by 4GL-type tools or non-IBM compilers, can contain null values for either the DB2 version, the host language, or both. When this occurs, the values that are specified on the DEFAULTS statement are inserted into the generated DBRMs.

The DEFAULTS command format is:

```
DEFAULTS DB2=n,LANGUAGE=lang
```

You can specify DB2, LANGUAGE, or both. Normally you should specify both to ensure that you provide defaults for all cases. Acceptable values for *n* and *lang* are shown in the following table.

Table 5. DEFAULTS command values

<i>n</i>	<i>lang</i>
3	ASSEMBLER
4	COBOL
5	COBOL_II
6	IBM_COBOL
7	FORTTRAN
8 and 9	PL/I
	C
	C++
	SQLJ

LIST

You can use the LIST command to obtain a report that identifies the DBRMs and packages that are present in the subsystem. This statement is optional. This command is normally used to determine which DBRMs or packages to generate using the DBRM and PACKAGE statements. Reports that are generated by the LIST command are written to *ddname* SYSPRINT.

The LIST command format is:

LIST *type name*

Specify either DBRM or PACKAGE for *type* and specify the name of the DBRM or package to list. Both operands are optional. If *name* is omitted, all DBRMs (or packages) in the catalog are listed. If both operands are omitted, all DBRMs and packages are listed, which can produce a very long report. If you specify LIST DBRM, each occurrence of the specified DBRM is listed along with the name of the plan into which it is bound. If you specify LIST PACKAGE, each occurrence of the specified package is listed along with its version and the name of the collection into which it is bound. You can use wild card characters to specify the name operand.

For example, if you specify LIST PACKAGE DSN*, the program will produce a report that lists all occurrences of all packages with names that begin with DSN.

PACKAGE

The PACKAGE command directs the program to re-create a single DBRM from a package using information from the DB2 catalog that is specified by the SSID statement. This statement is optional.

The PACKAGE command format is:

PACKAGE *collection.package.(version)*

Specify the collection name, the specific package within the collection, and the package version. All three qualifiers are required, as are parentheses around the version name. If a blank or null version is the target, indicate this by coding empty parentheses. For example, PACKAGE

collection.package.(). The generated DBRM is stored in the data set that is pointed to by ddname DBRMLIB under member name package. For example, the statement, PACKAGE DEMO.SAMPLE.(BACKUP) causes DBRM SAMPLE, version BACKUP, from collection DEMO to be re-created as member SAMPLE in the library that is referenced by DBRMLIB.

SSID

The SSID statement specifies the DB2 subsystem catalog where the DBRM or package information is to be obtained. This statement is required.

The SSID command format is:

```
SSID name/remotename
```

Specify the local subsystem ID and the remote subsystem ID of the DB2 subsystem catalog from which the DBRM is to be extracted. For example, SSID DB2A/DB64N specifies that DBRMs are to be extracted from local subsystem DB2A and remote subsystem DB64N.

The local SSID is required, the remote SSID is optional. If you do not specify a remote subsystem, data is retrieved from the local system.

Examples:

Example 1

Use the following control statements to list all DBRMs and packages with names that begin with DSN8 from subsystem TEST.

```
SSID TEST
LIST DBRM DSN8*
LIST PACKAGE DSN8*
```

Example 2

Use control statements similar to the ones in this example to generate one DBRM from a plan and one DBRM from a package by using default values for the DB2 version and host language.

```
SSID DB7A
DEFAULTS DB2=7,LANGUAGE=SQLJ
DBRM GL07P.GACP25
PACKAGE GLMSTR.GL4400.(GACP38)
```

Catalog timestamp listing (program BNDA030)

Use the BNDA030 batch program to identify all occurrences of a DBRM in the catalog and determine which instances have timestamps that match the timestamp in the DBRM.

With this program you can estimate which instances in the catalog are likely to be valid for use and which ones are possibly obsolete.

BNDA030 job control statements (JCL)

Various job control statements are associated with program BNDA030. Examples of these statements include EXEC, STEPLIB DD, and SYSOUT DD.

Member BNDJ030 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

Table 6. Job control statements for BNDA030

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. The format is: //stepname EXEC PGM=BNDA030 PGM=BNDA030 specifies that you want to run the timestamp list program, BNDA030.
STEPLIB DD	The STEPLIB DD statement defines an existing load library that contains the DBRM Checker programs. This DD statement is required unless BNDA030 was installed in the system link list. The DD statement must reference the load library that contains the DBRM Checker programs.
SYSOUT DD	The SYSOUT DD statement defines a sequential data set that is used for listing control statements and messages. BNDA030 writes a log of control statements and status messages to the SYSOUT DD statement. This DD statement is required. You can assign SYSOUT to SYSOUT or to any sequential data set.
SYSPRINT DD	The SYSPRINT DD statement defines a sequential data set that is used for output reports. Reports that are generated by the BNDA030 command are written to the SYSPRINT DD statement. This DD statement is required. You can assign SYSPRINT to SYSOUT or to any sequential data set.
DBRMLIB DD	The DBRMLIB DD statement identifies the partitioned data set that contains the DBRMs to be searched. Specific search targets are specified by the DBRM command. This statement is required.
SYSIN DD	The SYSIN DD statement defines the control data set. The statement must precede a valid set of BNDA030 commands. This statement is required.

BNDA030 commands

The BNDA030 program recognizes specific command statements that must adhere to defined syntax rules.

BNDA030 command statements are read from SYSIN. BNDA030 commands must adhere to the following syntax rules:

- Comment lines are indicated by an asterisk (*) in column one and are ignored.
- Comment lines are interspersed with command lines in any order. There is no limit to the number of comment lines allowed.
- Commands must be wholly contained between columns 1 and 72. Within this range, commands are free form and can start in any column.
- Each command must be contained on a single line. Continuing the command on a second line or multiple-line commands are not allowed.

Wild cards

Some commands permit the use of wild card characters to specify a range of one or more objects to be processed. The individual command descriptions indicate when you can use wild card characters. Where wild cards are allowed, the following conventions are supported:

- A percent sign (%) represents any single character.
- An asterisk (*) represents any number of characters.

Commands

The BNDA030 program recognizes the following commands:

DBRM

The DBRM command specifies the DBRMs to be compared to the catalog.

The DBRM command format is:

DBRM *name*

Specify the member name to be searched for in the DBRMLIB. You can specify the entire name or use wild card characters to search for patterns. For each DBRM match in DBRMLIB, the catalog is searched to locate all plans and packages that include it. A report is generated that includes instances with their timestamp. When the catalog timestamp matches the target DBRM timestamp, the report is flagged with the word MATCH. The report is written to SYSPRINT.

SSID

The SSID statement specifies the DB2 subsystem catalog to be searched for occurrences of the target DBRMs.

The SSID command format is:

SSID *name*

Specify the DB2 subsystem catalog name you want to search. For example, SSID DSN7 specifies that subsystem DSN7 is to be searched for target DBRMs.

Catalog cleanup (program BNDA036)

Program BNDA036, the DB2 Bind Manager catalog cleanup program that is used with DB2 Version 8 and later, correlates the package entries in the DBRM catalog to one or more load libraries to determine which packages are still referenced by load modules.

For each DBRM that is not referenced by a load module, the FREE command is generated and written to a sequential data set. You can issue the commands later to remove the orphan package entries from the catalog.

Attention: BNDA036 supports only DB2 Version 8 and later releases. To perform this function with DB2 Version 7 and earlier releases, use BNDA046.

This program performs the same basic function as program BNDA020 but imposes some restrictions that allow for faster processing with less overhead. There are two principal differences between this program and BNDA020:

- BNDA036 assumes that each package name matches the associated CSECT name (the CSECT name that corresponds to the object code of the program whose precompile generated the package).

Due to the inherently high resource consumption that is involved in this type of processing, the catalog clean up function runs as a batch job. The job is controlled by JCL.

BNDA036 does not process packages that are created by using the precompiler LEVEL option. Packages that are precompiled with the LEVEL option are flagged

in the SYSPRINT listing. When the step is completed, a message is issued that indicates the total number of packages that were bypassed.

BNDA036 job control statements

Various job control statements are associated with program BNDA036. Examples of these statements include EXEC, STEPLIB DD, and SYSOUT DD.

Member BNDJ036 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

Table 7. Job control statements for BNDA036

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. The format is: //stepname EXEC PGM=BNDA036 PGM=BNDA036 specifies that you want to run the catalog clean up program.
STEPLIB DD	The STEPLIB DD statement defines an existing load library that contains the DBRM Checker programs. This DD statement is required unless the DBRM Checker programs were installed in the system link list. The DD statement must reference the load library that contains the BNDA036 load module.
SYSPRINT DD	The SYSPRINT DD statement defines a sequential data set that is used for messages and output reports.
COMMANDS DD	The COMMANDS DD statement defines a sequential data set where the generated FREE commands will be written. The data set must have the attributes DSORG=PS, RECFM=FB, and LRECL=200.
SYSIN DD	The SYSIN DD statement defines the control data set. This statement must precede a set of valid BNDA036 commands. This statement is required.
LOAD nn DD	The LOAD nn DD statement defines an existing load library that contains load modules to be checked. The DD names must conform to the pattern LOAD nn where nn is exactly two numeric digits. Any library that is identified by such a DD statement is searched. You can specify the LOAD nn statements in any order and the numbers (nn) do not need to be consecutive or contiguous. Duplicates are not allowed, therefore, do not specify the same number twice. At least one LOAD nn statement is required.
SYSOUT DD	The SYSOUT DD statement defines a sequential data set that is used for sort output messages.
SW36IN DD	The SW36IN DD statement defines a sequential data set that is used as a temporary work file by the program.
SW36OUT DD	The SW36OUT DD statement defines a required sequential data set that is used as a temporary work file by the program.
SW36WK01 DD	The SW36WK01 DD statement defines a required sequential data set that is used as a temporary work file by the program.
SW36WK02 DD	The SW36WK02 DD statement defines a required sequential data set that is used as a temporary work file by the program.
SW36WK03 DD	The SW36WK03 DD statement defines a required sequential data set that is used as a temporary work file by the program.
BNDWK1 DD	The BNDWK1 DD statement defines a required sequential data set that is used as a temporary work file by the program.

Table 7. Job control statements for BNDA036 (continued)

Statement	Use
DBRMOUT DD	This DD statement is optional. If present, it defines a sequential data set that will contain backup copies of DBRMs that correspond to the packages that is being freed. For each FREE command that is generated by BNDA036, the corresponding DBRM will be written to this data set. The data set characteristics must include RECFM=FB and LRECL=80.
BNDCKPT DD	This DD statement is optional. However, it must be present in order to use the Checkpoint/Restart facility. It defines a sequential data set that is used to record checkpoint information. If present, it is written to during every run, and it is read from during a restart.

BNDA036 commands

The BNDA036 program recognizes specific command statements that conform to defined syntax rules.

BNDA036 command statements are read from SYSIN. BNDA036 commands conform to the following syntax rules:

- You must enter commands in the format `COMMAND(operand)`.
- Embedded blanks are not allowed.
- Each command must be entered on a separate line, between columns 1 and 72.

Wildcard characters

Some commands permit the use of wildcard characters to specify a range of one or more objects to be processed. The individual command descriptions indicate when you can use wild card characters. Where wild cards are allowed, the following conventions are supported:

- A percent sign (%) represents any single character.
- An asterisk (*) represents any number of characters.

Commands

The BNDA036 program recognizes the following commands:

COLLID

The COLLID statement specifies the collection IDs to be considered as potential candidates for freeing. This command is optional.

The COLLID command format is:

`COLLID name`

Specify the collection name. You can specify the entire name or use wild card characters to search for patterns. A FREE command is generated for any package that is contained in the target collections and is not matched by a load module.

You can issue this command up to a maximum of 25 times per run.

SSID

The SSID statement specifies the DB2 subsystem catalog where the DBRM or package information is to be obtained. This statement is required.

The SSID command format is:

SSID *name*

Specify the DB2 subsystem name where you want to search. A FREE command is generated for any package in this subsystem that is not matched by a load module.

You can issue this command only once per run.

PACKAGE

The PACKAGE command limits the number of packages processed in a particular execution. This can improve performance when processing a large catalog containing many hundreds or thousands of packages.

The PACKAGE command format is:

```
PACKAGE name1-name2
```

name1 is a required operand; *name2* is optional. If *name2* is not coded, omit the dash as well. No embedded blanks are allowed in the operand field. The first blank encountered terminates the operand field; the remainder of the line becomes a comment.

name1 (and optionally *name2*) are package (DBRM) names representing the lower and upper limits, respectively, of a range of packages to check. If only one name is specified then only that package is checked.

The PACKAGE command is optional. If it is present, only packages matching the specified names or ranges will be processed; no other packages will be scanned. A maximum of 25 PACKAGE commands may be specified at one time, i.e., in a single execution.

You can issue this command only once per run.

MODE

Dynamic Mode is activated by the MODE command. The MODE command is specified in the SYSIN input stream and has two forms:

```
MODE DYNAMIC=PROD  
MODE DYNAMIC=DEV
```

These two forms of the command invoke Dynamic Mode in production or development mode, respectively. The differences between production and development have to do mostly with how invalid and/or inoperative packages are handled. For a detailed description, see “Dynamic Mode for BNDA036” on page 55.

Specifying either production or development mode enables the use of dynamically allocated libraries.

MAXPKG

The MAXPKG command is part of the Checkpoint/Restart facility. MAXPKG limits the number of SYSPACKAGE catalog entries that will be tested in a single execution of BNDA036. This command limits run time by restricting the amount of data that is processed in a single job.

Important: When the Checkpoint/Restart facility is used, the analysis criteria may not be changed after the first job runs. Filtering and selection commands, such as MAXLM and MAXPG, can be coded only on the first job in a series; subsequent jobs must specify the RESTART command only.

The MAXPKG command is issued in the following format:

MAXPKG *n*

where *n* specifies the maximum number of packages to process. The *n* variable is required.

When *n* packages have been selected for testing, no more packages will be retrieved and scanned, regardless of the number of available catalog entries. For example, if MAXPKG 2000 is specified, either all of the packages that qualify or 2000 of them will be processed, whichever amount is less.

If a DBRM is bound multiple times, which can occur with multiple packages and collections, more entries for a given package might exist than are specified in MAXPKG. In this case, the program will terminate when MAXPKG is reached and all potential packages might not have been scanned. To avoid this case when you use MAXPKG, choose a number that is large enough.

MAXPKG can be used with the MAXLM command.

MAXLM

The MAXLM command is part of the Checkpoint/Restart facility. It limits the number of load modules that are processed during a single execution of BNDA036, and it controls run times by limiting the number of load modules to be scanned.

Attention: If you are not using the Checkpoint/Restart facility, which means you are submitting a single standalone job, do not issue the MAXLM command. The following example shows the effects of issuing the MAXLM command when you are submitting a standalone job: If 2000 load modules are in a load library and you specify that only 1000 are to be checked, some packages might be matched by load modules that were not checked. In this case, BNDA036 assumes that those packages have no matches, and FREE commands will be issued for them. MAXLM is designed to be used when a series of jobs that use RESTART is run. In this case, FREE commands will not be issued until the last job in the series has run and all load modules have been checked.

Important: When the Checkpoint/Restart facility is used, the analysis criteria may not be changed after the first job runs. Filtering and selection commands, such as MAXLM and MAXPG, can be coded only on the first job in a series; subsequent jobs must specify the RESTART command only.

The MAXLM command is issued in the following format:

MAXLM *n*

where *n* specifies the maximum number of load modules to scan in a single execution. The *n* variable is required.

When *n* load modules have been processed, no more load modules will be checked, regardless of the number or size of load libraries that are allocated. For example, if MAXLM 1000 is specified, either all of the remaining load modules or 1000 of them will be processed, whichever amount is less. A subsequent RESTART job will begin processing where this job leaves off and will process the next *n* load modules.

MAXLM can be used with the MAXPKG command.

RESTART

The RESTART command is part of the Checkpoint/Restart facility. RESTART causes execution to resume at a checkpoint. It must be specified by itself. It is not valid with any other commands. If you use the RESTART command, it must be the only command that you use.

The RESTART command is issued in the following format:

```
RESTART
```

The MAXPKG and MAXLM commands can be used to initiate a series of jobs that use the RESTART command. Values that are specified for MAXPKG and MAXLM remain in effect for all subsequent RESTART jobs until the series is processed completely.

If you are not using the Checkpoint/Restart facility, do not issue this command.

BNDA036 generates FREE commands only for programs coded in COBOL or PL/I; which languages are processed in a given execution depends on the particular combination of commands entered.

BNDA036 excludes from consideration any package or load module whose name begins with DSN, DGO, DSQ, SYS, or SQL. That is, it won't generate FREE commands for those modules. It similarly excludes packages bound remotely from another system, because such packages are incompletely represented in the local catalog (some columns are not populated, including the host language type, which is critical to BNDA036 processing). Except for these exclusions, BNDA036 will generate FREE commands for all non-matching packages, i.e., those for which it cannot find matching load modules.

Dynamic Mode for BNDA036

Dynamic mode is a special processing regime that changes BNDA036 processing in two ways.

Dynamic mode changes BNDA036 processing in two ways:

- Dynamic mode allows the load libraries being searched to be allocated dynamically, rather than being specified by JCL DD statements
- Dynamic mode invokes special, additional rules for determining which packages to release and which packages to retain

Dynamic Mode is specified by the MODE command; see “BNDA036 commands” on page 52 for details.

Allocating Libraries Dynamically

When Dynamic Mode is active, when the MODE command has been specified, the load libraries to be searched may be specified as a list of data set names, one per line, entered via the DYNLIBS DD statement. See “Sample JCL” on page 56 for an example.

Production Mode

When Production Mode is invoked (MODE DYNAMIC=PROD), the following rules are in effect:

- Packages marked INOPERATIVE in the catalog are listed in the SYSPRINT report, but are not freed.

- Packages not marked VALID in the catalog are listed in the SYSPRINT report, but are not freed.
- The newest version of each package will never be freed, regardless of all other considerations. In other words, at least one version of each package will always be retained.

Development Mode

When Development Mode is invoked (MODE DYNAMIC=DEV), the following rules are in effect:

- Packages marked INOPERATIVE in the catalog are listed in the SYSPRINT report and are freed.
- Packages not marked VALID in the catalog are retained if they are less than six months old; if more than six months old, they are freed.

Sample JCL

SAMPLIB member BNDJ36L contains a sample job using the Dynamic Mode features.

```

//BNDJ36L JOB (ACCT),' ',CLASS=?,MSGCLASS=?
//*
//* 5655-E43
//* (c) Copyright HLS Technologies, Inc. 2009 All Rights Reserved
//*
/*****
//*
//* BNDJ36L using dynamic allocation feature
//*
/*****
//*
//* Make the following changes before submitting this job:
//*
//* 1) Replace the string <DB2> with the high level qualifier
//*    of your DB2 libraries, e.g., DSN710, DSN810, etc.
//*
//* 2) Replace the string <WORK> with the high level qualifier
//*    to be used for the FREECMD data set created by this job.
//*    by this job
//*
//* 3) Replace the string <BNDLOAD> with the name of the
//*    Bind Manager load library
//*
//* 4) Replace the string <SSID> with the name of the
//*    subsystem for which you want to generate 'FREE' commands
//*
//* 5) Replace the LOAD00 DD statement with one or more
//*    LOADnn statements referencing the load libraries
//*    to be searched
//*
//* 6) If using the checkpoint/restart facility, change the
//*    data set name in the BNDCKPT DD to a valid name.
//*
/*****
/***** CHANGE LOG *****/
/*****
//*
//* AAKG-PK77089-24960
//* V2R4-FM10200-141751
//*
/*****
//*
//* BNDJ36L (catalog cleanup utility)
//*
//* Generates 'FREE' commands for unused packages
//*
/*****
//BND36L PROC DB2HLQ=, HLQ FOR DB2 LIBRARIES
// WKHLQ=, HLQ FOR WORK FILES
// LOADLIB= NAME OF DBRM CHECKER LOADLIB
//*
//* Delete old output data sets, if any
//*
//CLEAR EXEC PGM=IEFBR14
//BND36L DD DSN=&WKHLQ..FREECMD.DATA,
// DISP=(MOD,DELETE),
// UNIT=SYSALLDA,SPACE=(TRK,1),DSORG=PS

```

Figure 31. Sample job using the Dynamic Mode features (1 of 2)

```

//*
//* Generate 'FREE' commands
//*
//B36 EXEC PGM=BND A036,REGION=0M
//STEPLIB DD DSN=&LOADLIB,DISP=SHR
// DD DISP=SHR,DSN=&DB2HLQ..SDSNEXIT
// DD DISP=SHR,DSN=&DB2HLQ..SDSNLOAD
// DD DSN=CEE.SCEERUN,DISP=SHR
//COMMANDS DD DSN=&WKHLQ..FREECMD.DATA,DISP=(NEW,CATLG),
// UNIT=SYSALLDA,SPACE=(CYL,(5,5),RLSE),
// DSORG=PS,RECFM=FB,LRECL=200
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SW36IN DD UNIT=SYSALLDA,SPACE=(CYL,30)
//SW36OUT DD UNIT=SYSALLDA,SPACE=(CYL,30)
//SW36WK01 DD UNIT=SYSALLDA,SPACE=(CYL,30)
//SW36WK02 DD UNIT=SYSALLDA,SPACE=(CYL,30)
//SW36WK03 DD UNIT=SYSALLDA,SPACE=(CYL,30)
//BNDWK1 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//*
/** The following DD statement is optional. If it is present,
/** backup DBRMs will be written here for the packages to be
/** freed. Insert an appropriate data set name.
/**
/**DBRMOUT DD DISP=(NEW,CATLG),UNIT=SYSALLDA,SPACE=(CYL,(5,5)),
/** DSORG=PS,DSN=<dsname>
/**
/** The following two DD statements are optional. Uncomment one
/** or the other if you are using the checkpoint/restart facility.
/** (Use the first DD statement for the initial job and the second
/** for restarts.) Insert an appropriate data set name.
/**
/**BNDCKPT DD DISP=(NEW,CATLG),UNIT=SYSALLDA,SPACE=(CYL,5),
/** DSORG=PS,DSN=<dsname>
/**BNDCKPT DD DISP=OLD,DSN=<dsname>
/**
// PEND
/**
//RUN EXEC BND36,
// DB2HLQ=<DB2>, HLQ for DB2 libraries
// WKHLQ=<WORK>, HLQ for work files
// LOADLIB=<BNDLOAD>
//B36.LOAD00 DD DISP=SHR,DSN=<DB2>..SDSNLOAD *** SAMPLE ***
/**
/** Insert the names of the libraries to be allocated
/** dynamically after following the DYNLIBS DD statement.
/** 1 data set name per line, fully qualified, no quotes.
/**
//B36.DYNLIBS DD *
SAMPLE.DATASET.NAME
/**
//B36.SYSIN DD *

* Available modes are DYNAMIC=PROD and DYNAMIC=DEV

MODE DYNAMIC=PROD
SSID <SSID>
/**
//

```

Figure 32. Sample job using the Dynamic Mode features (2 of 2)

The Checkpoint/Restart facility

When the number of load modules to be scanned is large enough to extend BND A036 run times, the Checkpoint/Restart facility separates large runs into more manageable pieces.

The Checkpoint/Restart facility records information that describes the current status of a run in the checkpoint file (BNDCKPT DD). You can use the MAXPKG and MAXLM commands to initiate a series of jobs for the Checkpoint/Restart facility. When processing is complete for the MAXPKG command, the MAXLM command, or both, execution halts, and information is written to the checkpoint file, which indicates where the program stopped. Then, you specify only the RESTART command and resubmit the job. Resubmitting the job with only the RESTART command specified causes BND A036 to resume execution at the point where the previous job stopped. Continue submitting these restart jobs until all the

| data has been processed, at which point FREE commands are generated and
| DBRM backup copies are made, if they are requested. Values that are specified for
| MAXPKG and MAXLM remain in effect for all subsequent RESTART jobs until the
| series of jobs is processed completely.

| If load libraries are specified in the JCL, the LOAD nn DD statements must be the
| same for every run. If a restart job is submitted with a different set of LOAD nn DD
| statements than that of the original job, results are unpredictable. If dynamic load
| libraries are specified by using DYNLIBS, they need be specified only on the first
| initial job. They will be recorded in the checkpoint file and used for all subsequent
| jobs.

| For the first run, the checkpoint file listed in BNDCKPT DD must be a new data
| set, which will be initialized and written to. For all subsequent restart jobs, you
| must point to this data set so that restart information can be retrieved. When you
| use the Checkpoint/Restart facility, specify DISP=NEW on the first run, and
| specify DISP=OLD on subsequent runs.

| **Important:** When the Checkpoint/Restart facility is used, the analysis criteria may
| not be changed after the first job runs. Filtering and selection
| commands, such as MAXLM and MAXPG, can be coded only on the
| first job in a series; subsequent jobs must specify the RESTART
| command only.

Creating the index for BNDA036

You can improve the performance of BNDA036 by creating a special index before running the program.

To create the index for BNDA036:

1. Customize the JCL in sample member BNDJ36I. Sample member BNDJ36I is shown in the following two figures.

```

//BNDJ36I JOB (ACCT),' ',CLASS=?,MSGCLASS=?
/**
/** 5655-E43
/** (c) Copyright HLS Technologies, Inc. 2007 All Rights Reserved
/**
/*******
/**
/** This job creates an index for catalog table SYSIBM.SYSPACKAGE.
/**
/** When the DB2 catalog contains an exceptionally large number
/** of package entries, i.e., SYSIBM.SYSPACKAGE is very large,
/** this job may help to improve the performance of program
/** BNDA036
/**
/** To take advantage of this functionality, use the following
/** procedure:
/**
/** (1) Customize and submit this job. Verify that the job
/** runs successfully (condition code 0).
/**
/** (2) Bind program BNDA036. This is necessary to take
/** advantage of the new index created in step 1.
/**
/** (3) Run BNDA036.
/**
/** (4) Optional - if desired, delete (DROP) the index. The
/** index is named SYSIBM.SPKTEMP_IX.
/**
/*******
/**
/** Make the following changes before submitting this job:
/**
/** 1) Replace the string <DB2> with the high level qualifier
/** of your DB2 libraries, e.g., DSN710, DSN810, etc.
/**
/** 2) Replace the string <PLAN> with the name of the plan
/** for DSNTIAD on your system, e.g., DSNTIA71, DSNTIA81,
/** etc.
/**
/** 3) Replace the string <SSID> with the name of the
/** subsystem for which you want to generate 'FREE' commands
/**
/*******

```

Figure 33. Sample member BNDJ36I JCL (1 of 2)

```

/*
/***/ CREATE INDEX
/*
//CREATEX EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(<SSID>)
    RUN +
        PROGRAM(DSNTIAD) +
        PLAN(<PLAN>) +
        LIB('<DB2>.RUNLIB.LOAD')
/*
//SYSIN      DD *
CREATE INDEX SYSIBM.SPKTEMP_IX
ON SYSIBM.SYSPACKAGE
    (NAME,HOSTLANG,VERSION,COLLID,CONTOKEN)
USING      STOGROUP  SYSDEFLT
PRIQTY     100
SECQTY     50
PCTFREE    0
CLOSE      NO
;
/*
//

```

Figure 34. Sample member BNDJ36I JCL (2 of 2)

2. Bind program BNDA036 so that the program will recognize and use the new index.
3. Run BNDA036.
4. Optionally, DROP the index that you created in step one.

BNDA055

BNDA055 is designed to improve the performance of BNDA036 by removing the need for BNDA036 to scan large numbers of NCAL modules. Performance of BNDA036 can be improved by removing any libraries that consist exclusively of NCAL modules from the BNDA036 JCL and processing those libraries with BNDA055 instead.

Program BNDA055 filters the FREE commands that are generated by BNDA036 and removes FREE commands if both of the following conditions are true:

- An NCAL load module with a name that matches the package name exists.
- One or more packages with that name have been bound in the DB2 subsystem.

BNDA055 processes the FREE commands generated by BNDA036. Therefore, you must run BNDA055 after BNDA036, either as a later step in the same job that executes BNDA036 or as a separate job. In either case, the output file that is produced by BNDA036 must be input to BNDA055 (i.e., the FREECMD.DATA data set name for BNDA036 must be referenced in the //CMDIN DD statement for BNDA055).

Running BNDA055

Run BNDA055 after you run BNDA036 either in the same job stream or separately.

To run BNDA055:

1. Run the BNDBIND8 job or the BNDBIND job.
 - If you are using DB2 V8 or V9 or DB2 10 (CM), run BNDBIND8.
 - If you are using DB2 10 (NFM), run BNDBIND.
2. Run the BNDJ036 job.
3. Run the BNDJ055 job.

BNDA055 input and output

Program BNDA055 accepts specific types of input and generates several types of output.

Input

BNDA055 accepts the following types of input:

- Control statements.
- FREE commands that were previously generated by BNDA036.
- A list of data set names that identify the NCAL load modules to be scanned.

Output

BNDA055 generates the following types of output:

- Messages
- FREE commands (a filtered subset of the input)

BNDA055 job control statements

Based on the local site configuration and the JCL that is used to run BNDA036, the BNDA055 program might generate FREE commands that do not need to be run. BNDA055 reads the FREE commands that are produced by BNDA036, determines whether the two conditions are true, and, if both conditions are true, discards any FREE commands that do not need to be used.

All control statements are submitted through SYSIN.

BNDA055 uses the following statements:

Table 8. Job control statements for BNDA055

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement executes the program. Specify the EXEC statement in the following format: <i>//step-name</i> EXEC PGM=BNDA055,REGION=0M, where <i>step-name</i> is a job step name that you assign, PGM=BNDA055 specifies that you want to run the NCAL postprocessor program, and REGION=0M tells the operating system to allocate the maximum available memory to this job step.
STEPLIB	Load libraries. This concatenation must begin with the library that contains the BNDA055 program. Depending on the system configuration, it might need to include DB2 and Language Environment (LE) libraries.
SYSPRINT	The SYSPRINT DD statement defines a sequential data set that is used for messages.
SYSOUT	The SYSOUT DD statement defines a sequential data set that is used for error and diagnostic messages. Normally this data set is empty. If a failure or malfunction occurs, include the contents of SYSOUT as part of the problem description

Table 8. Job control statements for BNDA055 (continued)

Statement	Use
CMDIN	The CMDIN DD statement references a sequential data set or PDS member that contains the FREE commands that were previously generated by BNDA036.
CMDOUT	A sequential data set or PDS member to which BNDA055 will write the selected FREE commands (the contents of CMDIN less the records dropped, if any records exist, according to the criteria previously described).
SYSIN	The SYSIN DD statement defines the control data set. This statement must precede a set of valid BNDA055 commands. This statement is required.

BNDA055 recognizes specific control statements that adhere to a specific syntax. Control statements are read from SYSIN.

BNDA055 control statements must adhere to the following syntax:

- Control statements must be specified as **KEYWORD=operand**, where *operand* is a value that you specify.
- Each control statement must be specified on a separate line between columns 1 - 80.
- Control statements are free form and can begin in any column.
- Embedded blanks are not allowed.
- Any line that is entirely blank or that contains an asterisk (*) in column 1 is assumed to be a comment.

BNDA055 recognizes the following control statements:

SSID The SSID statement is required and may be issued only once per run. The statement uses the following syntax: **SSID=Subsystem_ID**, where *Subsystem_ID* is the subsystem whose catalog will be scanned. The following example shows the correct syntax: **SSID=DB8G**.

LOADLIB

The LOADLIB statement specifies the name of an NCAL load library to be scanned. You can specify 0 - 100 LOADLIB statements. The statement uses the following syntax: **LOADLIB=dsname**, where *dsname* is the load library name. The following example shows the correct syntax:
LOADLIB=ACCOUNT.NCAL.LOADLIB.

Sample BNDA055 job

Use the sample JCL as a guide to run BNDA055.

Replace the variables in **boldface** type with values that are appropriate for your environment. See SBNDSAMP(BNDJ055) for descriptions of these parameters. The sample JCL is included in SAMPLIB member BNDJ055.

```

//B55 EXEC PGM=BNDA055,REGION=0M
//STEPLIB DD DISP=SHR,DSN=<BNDLOAD>
// DD DISP=SHR,DSN=<DB2>.SDSNEXIT
// DD DISP=SHR,DSN=<DB2>.SDSNLOAD
// DD DISP=SHR,DSN=CEE.SCEERUN
//SYSPRINT DD SYSOUT=*
//CMDIN DD DISP=SHR,DSN=<OLDFREECOMMANDS>
//CMDOUT DD DISP=(NEW,CATLG),UNIT=SYSALLDA,SPACE=(CYL,(5,5),RLSE),
// DSN=<NEWFREECOMMANDS>
//SYSOUT DD SYSOUT=*
//SYSIN DD *
*
* EXACTLY ONE SSID STATEMENT MUST BE PRESENT.
* IT SPECIFIES THE SUBSYSTEM WHOSE CATALOG IS TO BE SCANNED.
*
* AT LEAST ONE LOADLIB STATEMENT MUST BE PRESENT.
* EACH LOADLIB STATEMENT IDENTIFIES ONE NCAL LIBRARY.
*
  SSID=<SSID>
  LOADLIB=<DSNAME>
/*

```

Figure 35. Sample BNDA055 job

Batch DBRM interpreter (BNDA043)

The batch DBRM interpreter program (BNDA043) is a batch program that is designed to produce the same output as the existing DBRM Browser ISPF function.

The output report that is produced by BNDA043 follows the same format as the output display that is produced by the ISPF-based browser. BNDA043 provides the capability to produce the output for many DBRMs in a single run, instead of one at a time as with the online function. BNDA043 can write the output to a data set where it can be saved.

JCL for sample BNDA043 output report

The following figure shows the JCL that produced the sample BNDA043 output report.

```

//DNS1B43 JOB (ACCT), 'DNS1',CLASS=A,MSGCLASS=X
//H43 EXEC PGM=BNDA043
//STEPLIB DD DISP=SHR,DSN=DEV.DV1.LOADLIB
// DD DISP=SHR,DSN=DSN810.SDSNEXIT
// DD DISP=SHR,DSN=DSN810.SDSNLOAD
//SYSUT1 DD DISP=(,DELETE),UNIT=SYSALLDA,SPACE=(TRK,75)
//DBRMLIB DD DISP=SHR,DSN=DSN710.SDSNDBRM
//SYSPRINT DD SYSOUT=*
//REPORT DD SYSOUT=*
//SYSIN DD *
  DISPLAY DBRM=DSN@CCOR
/*
//

```

Figure 36. JCL for sample BNDA043 output report

BNDA043 job control statements

Various job control statements are associated with program BNDA043, including REPORT DD, STEPLIB DD, and SYSPRINT DD.

The following DD statements are required by BNDA043.

Table 9. Job control statements for BNDA043

Statement	Use
STEPLIB	The STEPLIB DD statement identifies the load library that contains BNDA043. This statement is required unless the module is in the system link list.
SYSUT1	The SYSUT1 DD statement identifies a temporary work file that is used by BNDA043.
DBRMLIB	The DBRMLIB DD statement identifies a DBRM library. This DD statement is required if any DISPLAY DBRM commands are to be processed; otherwise, it is ignored.
SYSPRINT	The SYSPRINT DD statement specifies the destination for status messages that are produced by the program.
REPORT	The REPORT DD statement specifies the destination for the output report (interpreted DBRMs and packages).
SYSIN	The SYSIN DD statement identifies the source of the input commands. The source of the input commands can be either in-stream (DD *) or in a data set with attributes RECFM=FB and LRECL=80. See Table 12 on page 75, SYSUT1 DD for an example.

BNDA043 commands

The BNDA043 program recognizes specific command statements that conform to defined syntax rules.

You control BNDA043 batch processing with commands that you enter by using SYSIN. Three commands are available: SSID, DISPLAY DBRM, and DISPLAY PACKAGE.

SSID This command must be specified before any DISPLAY PACKAGE commands. It is ignored by DISPLAY DBRM commands.

The SSID command format is:

SSID *name*

where *name* is the name of the local DB2 subsystem to connect to.

DISPLAY DBRM

The DISPLAY DBRM command causes a DBRM to be retrieved from a DBRM library and displayed in interpreted form.

The DISPLAY DBRM command format is:

DISPLAY DBRM=*dbrmname*

where *dbrmname* is the name of a DBRM to be displayed. The DBRM is read from the DBRMLIB DD statement.

DISPLAY PACKAGE

The DISPLAY PACKAGE command causes a package DBRM to be retrieved from a DB2 subsystem catalog and displayed in interpreted form.

The DISPLAY PACKAGE command format is:

DISPLAY PACKAGE=*collection.package.(version)*

where *collection* is the name of the collection that contains the package to be displayed, *package* is the package name, and *version* is the package version ID. All three parameters are required. If the version name is blank, the parentheses must be coded as a placeholder, as in the following example:

```
DISPLAY PACKAGE=PAYROLL.PR700.()
```

The package is retrieved from the subsystem that is specified in the most recently encountered SSID command.

Example BNDA043 output report

The BNDA043 program produces a report that is based on the commands that you specify in the program JCL.

The following example shows a report that BNDA043 produces:

```
BNDA043      Contents Listing for DBRM DSN@CCOR in DSN710.SDSNDBRM
```

```
-----  
DBRM Header  
-----
```

```
Record ID (DBRMHID)          DBRM  
Record Length (DBRMHLEN)     160  
Precompile User ID (DBRMUSER) W98COMP  
Program name (DBRMPROG)      DSNACCOR  
Precompile timestamp (DBRMTIMS) 2004.10.11.22.34.22.541312  
                               (177E87B10A768014 hex)  
Options (DBRMPOPT)           APOST,PERIOD,APOSTSQL,GRAPHIC(NO),LANG=C,  
                               DECIMAL(15),NOFOLD,FLOAT(S390),SQL(DB2)  
Character set ID (DBRMCCSID)  0 (0000 hex)  
Max section number (DBRMMAXS) 0  
Entry stmt format (DBRMSTYPE) 1  
Reserved (DBRMSCC)           0000 (hex)  
Flags                         80 (hex)  
Dependency marker (DBRMPDRM)  N  
DB2 Version (DBRMMRIC)       K (Version 7)
```

```
-----  
Header Extension Record  
-----
```

```
Version                      UQ93893
```

```
-----  
SQL statements  
-----
```

```
Section 1  
Statement 788  
SQL statement:  
  DECLARE MSG_CSR CURSOR WITH RETURN WITH HOLD FOR SELECT RS_SEQUENCE , RS  
  _DATA FROM SESSION . DSNACCTMSG_TBL ORDER BY RS_SEQUENCE  
Host variables referenced by this statement:  
  (None)
```

```
Section 1  
Statement 1378  
SQL statement:  
  OPEN MSG_CSR  
Host variables referenced by this statement:  
  (None)
```

```
Section 2  
Statement 1481  
SQL statement:
```

```
DELETE FROM SESSION . DSNACCTMSG_TBL
Host variables referenced by this statement:
  (None)
```

```
Section 3
Statement 1498
SQL statement:
EXECUTE IMMEDIATE : H
Host variables referenced by this statement:
  TYPE=Input      stmt
```

```
Section 4
Statement 1797
SQL statement:
INSERT INTO SESSION . DSNACCTMSG_TBL ( RS_SEQUENCE , RS_DATA ) VALUES (
: H , : H )
Host variables referenced by this statement:
  TYPE=Input      rs_sequence
  TYPE=Input      rs_data
```

Catalog cleanup (program BNDA046)

Program BNDA046, the DB2 Bind Manager catalog clean up program that is used with DB2 Version 7 and earlier, correlates the package entries in the DB2 catalog to one or more load libraries to determine which packages are still referenced by load modules.

For each DBRM that is not referenced by a load module, the FREE command is generated and written to a sequential data set. You can issue the commands at a later time to remove the orphan package entries from the catalog.

Attention: BNDA046 supports only DB2 Version 7 and earlier. To perform this function with DB2 Version 8 and later, use BNDA036.

This program performs the same basic function as program BNDA020, but imposes some restrictions that allow for faster processing with less overhead. There are two principal differences between this program and BNDA020:

- BNDA046 assumes that each package name matches the associated CSECT name (the CSECT name that corresponds to the object code of the program whose precompile generated the package).

Due to the inherently high resource consumption that is involved in this type of processing, the catalog clean up function runs as a batch job. The job is controlled by JCL.

BNDA046 does not process packages that are created by using the precompiler LEVEL option. Packages that are precompiled with the LEVEL option are flagged in the SYSPRINT listing. When the step is completed, a message is issued that indicates the total number of packages that were bypassed.

BNDA046 job control statements

Various job control statements are associated with program BNDA046, including EXEC, STEPLIB DD, and SYSPRINT DD.

Member BNDJ036 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

Table 10. Job control statements for BNDA046

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. The format is: //stepname EXEC PGM=BNDA046 PGM=BNDA046 specifies that you want to run the catalog clean up program.
STEPLIB DD	The STEPLIB DD statement defines an existing load library that contains the DBRM Checker programs. This DD statement is required unless the DBRM Checker programs were installed in the system link list. The DD statement must reference the load library that contains the BNDA046 load module.
SYSPRINT DD	The SYSPRINT DD statement defines a sequential data set that is used for messages and output reports.
COMMANDS DD	The COMMANDS DD statement defines a sequential data set where the generated FREE commands will be written. The data set must have the attributes DSORG=PS, RECFM=FB, and LRECL=200.
SYSIN DD	The SYSIN DD statement defines the control data set. This statement must precede a set of valid BNDA046 commands. This statement is required.
LOAD nn DD	The LOAD nn DD statement defines an existing load library that contains load modules to be checked. The DD names must conform to the pattern LOAD nn where nn is exactly two numeric digits. Any library that is identified by such a DD statement is searched. You can specify the LOAD nn statements in any order and the numbers (nn) do not need to be consecutive or contiguous. Duplicates are not allowed, therefore, do not specify the same number twice. At least one LOAD nn statement is required.

BNDA046 commands

The BNDA046 program recognizes specific command statements that conform to defined syntax rules.

BNDA046 command statements are read from SYSIN. BNDA046 commands conform to the following syntax rules:

- You must enter commands in the format `COMMAND(operand)`.
- Embedded blanks are not allowed.
- Each command must be entered on a separate line, between columns 1 and 72.

Wildcard characters

Some commands permit the use of wildcard characters to specify a range of one or more objects to be processed. The individual command descriptions indicate when you can use wild card characters. Where wildcard characters are allowed, the following conventions are supported:

- A percent sign (%) represents any single character.
- An asterisk (*) represents any number of characters.

Commands

The BNDA046 program recognizes the following commands:

COLLID

The COLLID statement specifies the collection IDs to be considered as potential candidates for freeing. This command is optional.

The COLLID command format is:

COLLID *name*

Specify the collection name. You can specify the entire name or use wild card characters to search for patterns. A FREE command is generated for any package that is contained in the target collections and is not matched by a load module.

You can issue this command up to a maximum of 25 times per run.

SSID

The SSID statement specifies the DB2 subsystem catalog where the DBRM or package information is to be obtained. This statement is required.

The SSID command format is:

SSID *name*

Specify the DB2 subsystem name where you want to search. A FREE command is generated for any package in this subsystem that is not matched by a load module.

You can issue this command only once per run.

BNDA046 generates FREE commands only for programs coded in COBOL or PL/I; which language(s) are processed in a given execution depends on the particular combination of commands entered. Also, BNDA046 excludes from consideration any package whose name begins with DSN, DGO, DSQ, SYS, or SQL. That is, it won't generate FREE commands for those modules. It similarly excludes packages bound remotely from another system, because such packages are incompletely represented in the local catalog (some columns are not populated, including the host language type, which is critical to BNDA046 processing). Except for these exclusions, BNDA046 will generate FREE commands for all non-matching packages, i.e., those for which it cannot find matching load modules.

Creating the index for BNDA046

You can improve the performance of BNDA046 by creating a special index before running the program.

To create the index for BNDA046:

1. Customize the JCL in sample member BNDJ46I. Sample member BNDJ46I is shown in the following figure.

```

//BNDJ46I JOB (ACCT),' ',CLASS=?,MSGCLASS=?
//*
//* 5655-E43
//* (c) Copyright HLS Technologies, Inc. 2007 All Rights Reserved
//*
//*****
//*
//* This job creates an index for catalog table SYSIBM.SYSPACKAGE. *
//*
//* When the DB2 catalog contains an exceptionally large number *
//* of package entries, i.e., SYSIBM.SYSPACKAGE is very large, *
//* this job may help to improve the performance of program *
//* BNDA046. *
//*
//* To take advantage of this functionality, use the following *
//* procedure: *
//*
//* (1) Customize and submit this job. Verify that the job *
//* runs successfully (condition code 0). *
//*
//* (2) Bind program BNDA046. This is necessary to take *
//* advantage of the new index created in step 1. *
//*
//* (3) Run BNDA046. *
//*
//* (4) Optional - if desired, delete (DROP) the index. The *
//* index is named SYSIBM.SPKTEMP_IX. *
//*
//*****
//*
//* Make the following changes before submitting this job: *
//*
//* 1) Replace the string <DB2> with the high level qualifier *
//* of your DB2 libraries, e.g., DSN710, DSN810, etc. *
//*
//* 2) Replace the string <PLAN> with the name of the plan *
//* for DSNTIAD on your system, e.g., DSNTIA71, DSNTIA81, *
//* etc. *
//*
//* 3) Replace the string <SSID> with the name of the *
//* subsystem for which you want to generate 'FREE' commands *
//*
//*****

```

Figure 37. Sample member BNDJ46I JCL (1 of 2)

```

/*
**** CREATE INDEX
/*
//CREATX EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=<DB2>.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(<SSID>)
    RUN +
        PROGRAM(DSNTIAD) +
        PLAN(<PLAN>) +
        LIB('<DB2>.RUNLIB.LOAD')
/*
//SYSIN DD *
CREATE INDEX SYSIBM.SPKTEMP_IX
ON SYSIBM.SYSPACKAGE
(NAME,HOSTLANG,VERSION,COLLID,CONTOKEN)
USING STOGROUP SYSDEFLT
PRIQTY 100
SECQTY 50
PCTFREE 0
CLOSE NO
;
/*
//

```

Figure 38. Sample member BNDJ46I JCL (2 of 2)

2. Run the job.
3. Bind program BNDA046 so that the program will recognize and use the new index.
4. Run BNDA046.
5. Optionally, DROP the index that you created in step one.

Chapter 4. Bind Avoidance

When you use Bind Avoidance you can safely bypass the DB2 bind process for source code changes that do not alter the existing SQL structure. DB2 Bind Manager monitors the operation of the DB2 precompiler and determines if a bind is required by comparing the new and old DBRMs.

Access to the library that contains the DBRMs that were used during the previous bind operation is required. Bind Avoidance assumes that the information in the DBRM library is identical to the information that is used by DB2. Except when the optional DBRM Generation feature is used, Bind Avoidance does not connect to an active DB2 subsystem to validate DBRM and plan information.

To use DB2 Bind Manager, start the Bind Avoidance program, BNDAVB, instead of the standard DB2 precompiler. DB2 Bind Manager runs the precompiler for you and monitors its operation to determine if a bind is required. If a bind is required, the step condition code is set to 4. If you can safely skip the bind, the step condition code is set to 0 (zero).

Restriction: Bind Avoidance does not support plans or packages that are prepared by using the precompiler LEVEL option. The LEVEL option suppresses the generation of a unique consistency token. Because Bind Avoidance logic depends on the use of unique consistency tokens to make the bind or no bind decision, the LEVEL option renders Bind Avoidance inoperative.

You can run Bind Avoidance by using the following programs:

BNDAVB

For straightforward bind operations in which the precompile is followed by the compile and the bind.

BNDAVB1

For situations when the coprocessor is used.

BNDAVB3

For circumstances in which the following sequence of events occurs:

1. BNDAVB or BNDAVB1 is used on one subsystem.
2. The code is migrated through change control to another subsystem.
3. Another precompile does not occur, so only a bind is necessary.

JCL requirements for Bind Avoidance are the same as the requirements for the precompiler except that SYSPRINT must be allocated to a spool data set or a sequential data set. Use of a PDS member for SYSPRINT is not supported. To use Bind Avoidance, change the JCL EXEC statement to specify PGM=BNDAVB instead of PGM=DSNHPC.

Note: When coding JCL for program BNDAVB, SYSPRINT can be directed to a sequential data set or a member of a partitioned data set (PDS). When writing SYSPRINT to a PDS, always specify a member name in the JCL. If you specify a PDS with no member name, the data overwrites the PDS directory and corrupts the data set.

Topics:

- "Using BNDAVB"

Using BNDAVB

Use the Bind Avoidance program, BNDAVB, to determine if a bind is required.

Configuring coprocessor mode (program BNDAVB4)

With Bind Avoidance coprocessor support, Bind Avoidance functionality extends to operate in the coprocessor environment so that Bind Avoidance continues to be supported.

In the traditional operating environment, the DB2 precompiler runs as a separate job step, and DB2 Bind Manager intercepts the source code after the precompiler modifies it and before the language compiler sees it. Bind Avoidance operates on this intermediate source code before passing it to the compiler.

In a coprocessor environment, the precompiler and the language compiler are combined into a single step. Therefore, the intermediate source code is unavailable for manipulation.

Configuring for coprocessor support requires different JCL than operating Bind Avoidance in a traditional environment. The program name and the required DD statements are different. Coprocessor support adds two additional job steps: one before the compiler step and one after the compiler step.

Because of the methods that are used for compiler optimization, BNDAVB4 is required to run as a preprocessor to the present coprocessor. You must run the processes in the following order:

1. BNDAVB4. This program determines whether bind avoidance is possible.
2. The coprocessor. The coprocessor creates the executable and the DBRM.
3. BNDAVB1. This program post-processes the executable and the DBRM to complete the analysis and bypass the bind, if possible.

Important: BNDAVB4 applies only to COBOL.

BNDAVB4 job control statements

Various job control statements are associated with the BNDAVB4 catalog clean up program.

The following table shows job control statements for BNDAVB4.

Table 11. Job control statements for BNDAVB4

Statement	Use
JOB	The JOB statement starts the job.
EXEC	The EXEC statement starts the program. It is issued in the following format: //stepname EXEC PGM=BNDAVB4 where PGM=BNDAVB4 specifies that you want to run the bind avoidance coprocessor program, BNDAVB4.

Table 11. Job control statements for BNDAVB4 (continued)

Statement	Use
STEPLIB DD	<p>The STEPLIB DD statement defines an existing load library that contains the DB2 Bind Manager programs.</p> <p>This DD statement is required unless BNDAVB4 was installed in the system link list. The DD statement must reference the load library that contains the DB2 Bind Manager programs.</p>
SYSPRINT DD	<p>The SYSPRINT DD statement defines a sequential data set that is used for listing control statements and messages. BNDAVB4 writes a log of control statements and status messages to the SYSPRINT DD statement. This DD statement is required. You can assign SYSPRINT to SYSOUT or to any sequential data set.</p>
DBRMIN DD	<p>The DBRMIN DD statement defines an existing data set that contains the preexisting DBRM (the DBRM that existed before the compile step was run). This DD statement must reference a PDS member with attributes RECFM=FB and LRECL=80.</p>
SOURCEIN DD	<p>The SOURCEIN DD statement defines an existing data set that contains the preexisting COBOL source that is to be compiled.</p>
SOURCEOT DD	<p>The SOURCEOT DD statement defines the data set that is to be sent to the coprocessor for compilation.</p>

BNDAVB1 job control statements

Various job control statements are associated with the BNDAVB1 analysis.

The following table shows job control statements for BNDAVB1.

Table 12. Job control statements for BNDAVB1

Statement	Use
EXEC	<p>The EXEC statement starts the program. It is issued in the following format:</p> <pre>//stepname EXEC PGM=BNDAVB1</pre> <p>where PGM=BNDAVB1 specifies that you want to run the bind avoidance coprocessor program, BNDAVB1.</p>
STEPLIB DD	<p>The STEPLIB DD statement defines an existing load library that contains the DB2 Bind Manager programs.</p> <p>This DD statement is required unless BNDAVB1 was installed in the system link list. The DD statement must reference the load library that contains the DB2 Bind Manager programs.</p>
SYSPRINT DD	<p>The SYSPRINT DD statement defines a sequential data set that is used for listing control statements and messages. BNDAVB1 writes a log of control statements and status messages to the SYSPRINT DD statement. This DD statement is required. You can assign SYSPRINT to SYSOUT or to any sequential data set.</p>
OLDDBRM DD	<p>The OLDDBRM DD statement defines an existing data set that contains the preexisting DBRM (the DBRM that existed before the compile step was run). This DD statement must reference a sequential data set or a PDS member with attributes RECFM=FB and LRECL=80.</p>
NEWDBRM DD	<p>The NEWDBRM DD statement defines an existing data set that contains the new DBRM that was generated by the compile step. This DD statement must reference a PDS member with attributes RECFM=FB and LRECL=80.</p>

Table 12. Job control statements for BNDAVB1 (continued)

Statement	Use
OBJECT DD	The OBJECT DD statement defines an existing data set that contains the object deck that was generated by the compile step. This DD statement must reference a sequential data set or a PDS member with attributes RECFM=FB and LRECL=80.
SYSUT1 DD	The SYSUT1 DD statement defines a temporary work file that is used by BNDAVB1. This DD statement must reference a sequential data set with attributes RECFM=FB and LRECL=80. SYSUT1 must be allocated to a temporary data set and not kept after the BNDAVB1 step runs.

The following example shows DB2 Bind Manager used with an integrated coprocessor. The coprocessor integrates the DB2 precompiler with the language compilers. This example shows a COBOL compilation with the coprocessor and DB2 Bind Manager. The following example shows examples of job control statements for running BNDAVB4 and BNDAVB1.

```

//BNDAVB1 JOB (ACCT),' ',CLASS=?,MSGCLASS=?
//*
//*****
//*
//* Before executing this job:
//*
//* (1) Substitute a valid job card for the template above
//*
//* (2) In the JCL, replace all lower case items with values
//* that are appropriate for the local environment
//*
//*****
//***** CHANGE LOG *****
//*****
//*
//* AAMV 20110522 WNFR
//*
//*-----+
//*
//* BNDAVB1 Bind Avoidance (coprocessor mode)
//*
//*****
//*-----+
//*
//* Save a working copy of the old DBRM
//*
//*-----+
//*
//SAVE EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,DSN=existing.dbrmlib(dbrmname)
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// LRECL=80,RECFM=FB,
// DISP=(,PASS),DSN=&&OLDDBRM
//*
//*-----+
//*
//* Execute BNDAVB4 to prepare the program for compilation
//*
//*-----+
//*
//STEP1 EXEC PGM=BNDAVB4
//STEPLIB DD DISP=SHR,DSN=bnd.loadlib
//SYSPRINT DD SYSOUT=*
//DBRMIN DD DISP=SHR,DSN=existing.dbrmlib(dbrmname)
//SOURCEIN DD DISP=SHR,DSN=source.program
//SOURCEOT DD UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// LRECL=80,RECFM=FB,
// DISP=(,PASS),DSN=&&SRC
//*
//*-----+
//*
//* Compile the target program
//*
//*-----+
//*
//COB EXEC PGM=IGYCRCTL,
// REGION=4M,
// PARM=SQL
//STEPLIB DD DISP=SHR,DSN=IGYxxx.SIGYCOMP
// DD DISP=SHR,DSN=DSNxxx.SDSNEXIT

```

Figure 39. Coprocessor support sample JCL (Part 1)

```

//          DD DISP=SHR,DSN=DSNxxx.SDSNLOAD
//SYSLIB   DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSTEM  DD SYSOUT=*
//SYSUT1  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT2  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT3  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT4  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT5  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT6  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSUT7  DD UNIT=SYSALLDA,SPACE=(CYL,10)
//SYSLIN  DD UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//          LRECL=80,RECFM=FB,
//          DISP=(,PASS),DSN=&&OBJECT
//DBRMLIB DD DISP=OLD,DSN=dbrmlib.data.set.name(member)
//SYSIN   DD DISP=(OLD,DELETE),DSN=&&SRC
//*
/*-----+
/*
/* Execute BNDAVB1 to determine whether binding is required
/*
/*-----+
/*
//BNDAVB EXEC PGM=BNDAVB1
//STEPLIB DD DISP=SHR,DSN=bnd.loadlib
//SYSPRINT DD SYSOUT=*
//OLDDBRM DD DISP=(OLD,DELETE),DSN=&&OLDDBRM
//NEWDBRM DD DISP=OLD,DSN=dbrmlib.data.set.name(member)
//OBJECT DD DISP=(OLD,PASS),DSN=&&OBJECT
//SYSUT1 DD DISP=(NEW,DELETE),UNIT=SYSALLDA,SPACE=(CYL,1),
//          DSORG=PS,RECFM=FB,LRECL=80
/*
/*
/*-----+
/*
/* Bind the program if necessary
/*
/*-----+
/*
//DB2BIND EXEC PGM=IKJEFT01,COND=(0,EQ,BNDAVB)
//STEPLIB DD DISP=SHR,DSN=DSNxxx.SDSNLOAD
//DBRMLIB DD DISP=SHR,DSN=dbrmlib.data.set.name(member)
//SYSTSPT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
          Bind commands here
/*
//

```

Figure 40. Coprocessor support sample JCL (Part 2)

Using BNDAVB3 to eliminate BIND processing after a precompile

Use the BNDAVB3 program to avoid BIND processing after a successful BNDAVB precompile.

Prerequisite: To use BNDAVB3, you must have already run BNDAVB.

Use program BNDAVB3 to install a program into a DB2 subsystem if you do not have access to the results of the precompile process. BNDAVB3 is used at a later point in time than BNDAVB, when the results from BNDAVB are no longer available. BNDAVB3 identifies the program (DBRM) that is being installed from the normal BIND commands. BNDAVB3 also checks the DB2 catalog

(SYSIBM.SYSPACKAGE) to determine if the program has already been bound to that collection with the same program name (DBRM) and contoken value.

Programs that require BIND processing have the BIND command written to BINDOUT. If none of the bind commands need to be processed, the return code is set to 4. The only contents of BINDOUT are the DSN command and END. If the return code was set to 0, at least one of the BIND commands must be processed. If the BIND commands include a BIND PLAN with PKLIST, the command is always passed through to be processed. If the DBRM does not exist in the library that is specified with DBRMIN, the BIND commands are passed to BINDOUT. The return code is set to 0.

BNDAVB3 job control statements (JCL)

Various job control statements are associated with program BNDAVB3, including STEPLIB DD, BINDOUT DD, and SYSPRINT DD.

Member BNDAVB3 in the SBNDSAMP product sample library contains a sample job that illustrates how to compose the JCL.

The following table shows job control statements for BNDAVB3.

Table 13. Job control statements for BNDAVB3

Statement	Use
STEPLIB DD	The load library for BIND Manager and a current DB2 SDSNLOAD.
BINDOUT DD	Bind commands to be processed (return code = 0).
BINDREJ DD	Optional ddname for the BIND commands that are not required.
SYSPRINT DD	Optional ddname for any error messages and program identification.
DBRMIN DD	The DBRMLIB data set that contains the DBRMs to be processed.
BINDIN DD	The bind commands with the included DSN command. The DSN command is used to identify which DB2 subsystem to check for existing entries in SYSIBM.SYSPACKAGE

Generating SMF records

DB2 Bind Manager can create SMF records that document the results of the bind decision. You can use this information to gauge the ratio of bind-to-no-bind decisions and for workload measurement and projections.

This feature applies only to BNDAVB.

To generate SMF records:

Add a DD statement in the following format to the execution JCL:

```
//SMFSMF record number DD DUMMY
```

where the *SMF record number* (type) must be 128 - 255. Numbers outside this range are reserved by IBM. If you use this feature, the Bind Avoidance module must run from an APF-authorized library (authorization is required to cut SMF records). The following table shows the SMF record format.

Table 14. DB2 Bind Manager SMF record

Offset (Decimal)	Field name	Length (Bytes)	Remarks
18	JOBNAME	8	
26	DBRMNAME	8	
34	USERID	7	
41	AVBFLAG	1	'Y' = DB2 Bind Manager Successful (no bind required) 'N' = DB2 Bind Manager Unsuccessful (bind required)
42	TIMESTAMP	26	Old timestamp value (meaningful only if AVBFLAG='Y')

Related tasks:

“Step 5: Copy the BNDAVB module to an authorized library” on page 20

Related information:

 IBM Publications Center

Changing the report destination

DB2 Bind Manager usually issues its report on SYSPRINT, which it shares with the precompiler. However, you can direct the DB2 Bind Manager report to a different ddname.

This feature applies only to BNDAVB.

To separate the BNDAVB and precompiler output to change the report destination:

Add the BNDLOG DD statement to the JCL, as shown in the following example:

```
//BNDLOG DD SYSOUT=*
```

BNDAVB writes its report and messages to BNDLOG instead of to SYSPRINT.

Checking for precompiler warnings

By default, DB2 Bind Manager does not react to warning-level diagnostic messages that are issued by the DB2 precompiler. However, you can set DB2 Bind Manager to detect and set a return code for precompiler warnings messages.

This feature applies only to BNDAVB.

After DB2 Bind Manager determines that a bind is not required, a 0 return code is issued to control processing of subsequent job steps. A 0 return code is issued even when the precompiler return code is 4. A 4 return code indicates the presence of a warning-level diagnostic.

To direct DB2 Bind Manager to detect and set a return code for precompiler warning messages:

Add a dummy DD statement that uses SETRC n as the ddname in the DB2 Bind Manager JCL. In the ddname, n specifies the return code that DB2 Bind Manager

issues when a warning-level diagnostic from the precompiler is detected. For example, the following DD statement causes DB2 Bind Manager to return a condition code of 2 when DSNHPC issues one or more warning-level diagnostic messages.

```
//SETRC2 DD DUMMY
```

Possible *n* values are 2, 4, 6, and 8.

Activating DBRM generation

DB2 Bind Manager compares the results of the current precompile (the *new* DBRM) to the results of the previous precompile (the *old* DBRM). If the previous precompile is unavailable, the DBRM generates a temporary copy to satisfy the requirement.

This feature applies only to BNDAVB.

During precompiling, the old DBRM must be available when you compare the current and the previous precompiles. However, if it is not the policy at your installation to save DBRMs, the old DBRM is not available for comparison. DBRM generation circumvents this limitation by generating a temporary working copy of the old DBRM from information in the DB2 catalog which allows the comparison to proceed without interruption. DBRM generation deletes the temporary DBRM when the job step ends.

To activate DBRM generation

Add a dummy DD statement to the execution JCL. Preface the ddname with SSID to signal to DB2 Bind Manager that DBRM generation is in effect. The remainder of the ddname (1 - 4 characters) specifies the DB2 subsystem name where DBRM information is to be extracted. In the following example, DBRM generation is invoked from subsystem DSN1.

```
//SSIDDSN1 DD DUMMY
```

Ignoring DECLARE TABLE statements

Invoking the DECLARE TABLE exception causes DB2 Bind Manager to ignore DECLARE TABLE statements when it compares the old and new DBRMs.

This feature applies only to BNDAVB.

To ignore DECLARE TABLE statements:

Add the SKIPDECL DD statement to the execution JCL, as shown in the following example:

```
//SKIPDECL DD DUMMY
```

Suppressing CCSID checking (NOCC)

Bind Avoidance performs many checks, one of which compares the character set IDs (CCSIDs) of the old DBRMs against new DBRMs. If they do not agree, Bind Avoidance considers the comparison a mismatch and requires a bind.

This feature applies only to BNDAVB.

To suppress CCSID checking (NOCC):

| Add the NOCC DD statement to the execution JCL, as shown in the following
| example:

| //NOCC DD DUMMY

|

Chapter 5. Troubleshooting

Use these topics to diagnose and correct problems that you experience with DB2 Bind Manager

Topics:

- “DB2 Bind Manager Messages and codes” on page 84
- “Gathering diagnostic information”

Gathering diagnostic information

Before you report a problem with DB2 Bind Manager to IBM Software Support, you need to gather the appropriate diagnostic information.

Provide the following information for all DB2 Bind Manager problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version and mode of DB2 and z/OS that you are using
- The type and version of the operating system that you are using

Provide the following information for all DB2 Bind Manager batch problems:

- A complete joblog with msglevel (1) showing the error
- A dump of the abend and the complete joblog
- A complete joblog showing the loop (ex. a message is repeated multiple times)
- A complete joblog and a dump while in the hung condition
- A complete joblog and a sample of the bad output

Provide the following information for all DB2 Bind Manager interactive problems:

- Screen prints showing the steps that occurred before the failure, as possible

Other information to gather:

- DBRM

The DBRM is a unique file. You must use the TSO SXMITS command to transform the DBRM into a .bin file so that it can be received at IBM in a readable format. To format the file correctly and transmit it, complete the following steps:

1. Issue the TSO SXMITS command: XMIT NODE.user
DSN ('DBRM NAME')
MEMBER (NAME)
OUTDSN (OUT.DS.NAME.BIN)
SEQ
2. Save the file on the hard drive as a .bin file. Ensure the file always remains a .bin file; otherwise, the file will be unreadable when it arrives at IBM.
3. FTP the .bin file to IBM.
IBM will perform the RECEIVE to complete the process.

DB2 Bind Manager Messages and codes

These topics describe the error messages and codes that are issued by DB2 Bind Manager.

All DB2 Bind Manager and DBRM Checker messages and diagnostics are sent to SYSPRINT.

DB2 Bind Manager Return codes

These topics describe the return codes when a module completes a process. Refer to the corresponding table for the module you are running for a description of the return code.

DB2 Bind Manager return codes

On normal completion of DB2 Bind Manager processing, one of the codes described in the following table is returned. The codes are the same whether you are using traditional DB2 Bind Manager (program BNDAVB) or coprocessor DB2 Bind Manager (program BNDAVB1).

Table 15. DB2 Bind Manager return codes

Return Code	Meaning
0	DB2 Bind Manager has completed successfully. The new source has passed SQL validation checks and a DB2 bind is not required.
4	DB2 Bind Manager has determined that either the SQL validation has failed or an error has occurred. Diagnostic messages have been issued. A DB2 bind is required.
16	DB2 Bind Manager has encountered a severe error and cannot make a bind or no bind determination. This return code is accompanied by a message describing the specific error.

DBRM Cross-reference return codes

On normal completion of DBRM cross-reference processing, one of the codes described in the following table is returned.

Table 16. DBRM Cross-Reference return codes

Return Code	Meaning
0	DBRM Cross-Reference has completed successfully.
4	There is an error in the input syntax.
12	Fatal error. Contact your IBM representative.

DBRM Checker return codes

On normal completion of DBRM Checker processing, one of the codes described in the following table is returned.

Table 17. DBRM Checker return codes

Return Code	Meaning
0	DBRM Checker processing has completed successfully. The DBRMs pointed to by the DBRMLIB DD statement are consistent with the DB2 Catalog tables. A DB2 bind is not required.

Table 17. DBRM Checker return codes (continued)

Return Code	Meaning
4	DBRM Checker has determined that either the DBRMs did not match the DB2 catalog tables or the plan was not found. A DB2 bind is required.

BNDAVB3 return codes

On normal completion of BNDAVB3 program processing, one of the codes described in the following table is returned.

Table 18. BNDAVB3 return codes

Return Code	Meaning
0	At least one BIND command needs to be processed.
4	No bind commands need to be processed - skip the bind step 16.
16	Fatal error - see SYSPRINT for error messages. Unable to connect to DB2 or missing ddname.

BNDAVB4 return codes

On normal completion of BNDAVB4 program processing, one of the following codes is returned:

Table 19. BNDAVB4 return codes

Return Code	Meaning
0	Processing is complete.
16	Processing failed. Review the JCL and the output.

DB2 Bind Manager Message severities

Numbered messages follow the form BND $nnmt$, where nnn is the message number and t is a one-character suffix that indicates the type and severity of the message. The following table shows the suffix values and their meanings.

Table 20. DBRM Checker message severities

Suffix	Value
I (Informational)	Routine notification; no action required.
W (Warning)	Notifies the user of a possibly anomalous circumstance that might or might not reflect an error.
E (Error)	Processing cannot continue for the reason given in the message; probable user error.
S (Severe)	Environmental problem or logic error; continued processing is impossible.

DB2 Bind Manager will make every attempt to recover from most error situations. Under extreme conditions, such as hardware and detected logic errors, the program will abnormally end (ABEND). In these situations, the old DBRM library should be validated before restarting the job.

DB2 Bind Manager messages

DB2 Bind Manager messages conform to defined syntax rules

Messages are of the form BND $nnnt$, where nnn is the message number and t is a one-character suffix with the value I for informational messages, E for errors, and S for severe errors.

BND001I BIND MANAGER STARTED FOR DBRM=*dbrmname*

Explanation: Issued during DB2 Bind Manager initialization.

Programmer response: None.

Module: BNDAVB

BND002I OLD DBRM INFORMATION:TIMESTAMP:CHAR=*yyyy.mm.dd.hh.mm.ss.xxxx* HEX=*hhhhhhhh* *hhhhhhhh* DECIMAL=*ddddddd dddddd* USERID=*userid* VERSION=*version* CCSID=*ccsid*

Explanation: Issued after the old (preexisting) DBRM has been located and successfully processed. The timestamp in formation is displayed in character, hexadecimal (h) and decimal (d) formats, along with the ID of the last user to precompile the program. The VERSION information is included in the message only if the DBRM was created with the VERSION parameter. For DBRMs that were created before DB2 Version 8 new-function mode, the CCSID will always be zero.

Programmer response: None.

Module: BNDAVB

BND003I NEW DBRM INFORMATION:TIMESTAMP:CHAR=*yyyy.mm.dd.hh.mm.ss.xxxx* HEX=*hhhhhhhh* *hhhhhhhh* DECIMAL=*ddddddd dddddd* USERID=*userid* VERSION=*version* CCSID=*ccsid*

Explanation: Issued after a new DBRM was successfully created by the DB2 precompiler. This message is similar to BND002I, but shows the newly created timestamp and the ID of the current user, i.e., the user running the job. The VERSION information is included in the message only if the DBRM was created with the VERSION parameter. For DBRMs that were created before DB2 Version 8 new-function mode, the CCSID will always be zero.

Programmer response: None.

Module: BNDAVB

BND004I PROCESSING COMPLETED SUCCESSFULLY, * BIND NOT REQUIRED *****

Explanation: Issued at end-of-job when DB2 Bind Manager determines that a bind is not required. The old DBRM and subsequent source updates are compatible. This message is accompanied by condition code 0. You can process the load module without a DB2 bind.

Programmer response: None, if the bind step was changed to test for a return code from DB2 Bind Manager. Otherwise, omit binding the new DBRM.

Module: BNDAVB

BND005I DBRM COMPARE FAILED, SOURCE MODIFICATIONS HAVE CHANGED THE CONTENT OF THE DBRM

Explanation: Updates to the source code have resulted in a change to the SQL structure. A DB2 bind is required. This message is followed by a dump of both the old and the new DBRM records that triggered the mismatch.

Programmer response: None, if the bind step has been changed to test for a return code from DB2 Bind Manager. Otherwise, bind the new DBRM and copy it to the DBRMLIB.

Module: BNDAVB

BND006E *ddname* DCB FAILED TO OPEN

Explanation: DB2 Bind Manager was unable to open the *ddname* *ddname*.

Programmer response: Correct the DD specification and resubmit the job.

Module: BNDAVB

BND007E *sss* TIME STAMP HAS INVALID FORMAT, "LEVEL" PRECOMPILER OPTION NOT SUPPORTED

Explanation: The old or new (*sss*) DBRM was generated using the DB2 precompiler LEVEL option. This option is not supported by DB2 Bind Manager.

Programmer response: If *sss* is OLD then delete the old DBRM from library pointed to by the DBRMLIB DD. If *sss* is NEW then remove the LEVEL option from the PARM= string on the EXEC statement.

Module: BNDAVB

BND008E LANGUAGE TYPE *type* IS NOT SUPPORTED

Explanation: The HOST precompiler parameter specified a language, *type*, that is not supported by DB2 Bind Manager. Supported language types are COB2, COBOL, PLI, and ASM.

Programmer response: Correct the parameter value and resubmit the job.

Module: BNDAVB

BND010S INTERNAL ERROR DURING DECODE OR UPDATE OF THE NEW DBRM

Explanation: DB2 Bind Manager has encountered an internal processing error during DBRM update. The contents of the new DBRM are unpredictable.

Programmer response: Verify that DBRMLIB specifies a valid library and member name and that the old DBRM contains valid data. If this does not resolve the problem, save the SYSUDUMP and contact IBM Software Support.

Module: BNDAVB

BND011E DBRM EXCEEDS THE 3,000 SQL CALL RESTRICTION

Explanation: DB2 Bind Manager has detected a DBRM that contains more than 3,000 SQL calls. This exceeds the maximum supported program size.

Programmer response: Restructure the application to reduce the number of SQL calls.

Module: BNDAVB

BND012E SYNTAX ERROR OR UNRECOGNIZABLE OPERAND IN THE PRECOMPILER PARAMETER

Explanation: DB2 Bind Manager has detected a syntax error in the DB2 precompiler PARM= string.

Programmer response: Correct the error and resubmit the job.

Module: BNDAVB

BND013E UNABLE TO ALLOCATE THE TEMPORARY DBRM SAVE DATASET

Explanation: DB2 Bind Manager was unable to allocate a temporary DASD data set for its work file.

Programmer response: Ensure that sufficient DASD work space is available and resubmit the job.

Module: BNDAVB

BND014E RETURN CODE FROM THE SQL PRECOMPILER > 4

Explanation: The DB2 precompiler ended with a return code greater than four.

Programmer response: Correct the SQL errors in the source code and resubmit the job.

Module: BNDAVB

BND015E DBRMLIB DATASET HAS AN LRECL OTHER THAN 80

Explanation: DB2 Bind Manager and the DB2 precompiler require a record length of eighty bytes (LRECL=80) for the data set specified in the DBRMLIB DD statement.

Programmer response: Verify that DBRMLIB points to the correct old DBRM and that the library has the correct attributes (RECFM=FB and LRECL=80), and resubmit the job. If this does not resolve the problem, contact IBM Software Support.

Module: BNDAVB

BND016E MEMBER POINTED TO BY THE DBRMLIB DD IS NOT A VALID DBRM

Explanation: DB2 Bind Manager has detected an invalid DBRM. This message usually indicates that the DBRMLIB DD statement does not specify a valid DBRMLIB library.

Programmer response: Ensure that the library referenced by this ddname is the correct DBRM library.

Module: BNDAVB

BND017E UNABLE TO FIND OLD DBRM MEMBER IN DBRMLIB, *dbrmname* IS ASSUMED TO BE NEW

Explanation: DB2 Bind Manager could not locate the old DBRM member in the library pointed to by the DBRMLIB DD statement. DB2 Bind Manager processing is bypassed for this DBRM.

Programmer response: Ensure that the library referenced by this ddname is the correct DBRM library.

Module: BNDAVB

BND018S BIND MANAGER HAS DETECTED AN Sxxx ABEND, ATTEMPTING RECOVERY

Explanation: An ABEND has occurred during processing. DB2 Bind Manager's ESTAE routine is attempting to recover and continue.

Programmer response: None if recovery is successful

BND019E • BND025I

and the job completes normally. Otherwise, contact IBM Software Support

Module: BNDAVB

BND019E **Deletion of source statements has invalidated the DBRM—add comment lines to correct // BND019E Old stmt nbr=xxxx (hex), New stmt nbr=yyyy (hex)**

Explanation: This condition is caused by deleting statements from a large source module. Because of a limitation of the DB2 precompiler, the internal format of the DBRM for a COBOL program changes when the growth of a source program forces at least one SQL statement to have a relative line number greater than 9999. This does not pose a problem when it occurs, but it does cause a downward compatibility problem later if the lines are subsequently deleted, causing SQL statements to again have line numbers less than 10,000. This sequence of events creates a condition DB2 Bind Manager cannot handle, so the bind is not avoidable in such a case.

The condition can exist only for DB2 release 7 and prior, or release 8 in compatibility mode. The situation does not arise in release 8 new function mode or beyond.

In the message text, *xxxx* and *yyyy* are the statement numbers from the DBRM records in which the condition was detected.

Programmer response: If the SQL structure has not been modified, you can circumvent this situation by doing the following steps:

1. Restore the Δ old Δ DBRM (the version that was saved in the job step preceding the precompiler step).
2. Rather than deleting source statements, comment them out. This will leave the relative line numbers unchanged.
3. Rerun the compile job.

Module: BNDAVB

BND020E **DB2 VERSION/RELEASE MISMATCH BETWEEN OLD/NEW DBRMS**

Explanation: The old and new DBRMs were created under different releases of DB2.

Programmer response: An initial bind must be done under the new release; afterwards DB2 Bind Manager will function normally with the new DBRM format. To allow for possible modifications to the optimizer, it is always best to bind existing applications when the DB2 version or release level changes.

Module: BNDAVB

BND022E **DBRM COMPARE FAILED—NON-MATCHING CHARACTER SETS (CCSIDs)**

Explanation: A mismatch was detected between the old and the new DBRMs due to the fact that they were created with different character sets (CCSIDs). A bind is required.

Programmer response: None, if the bind step has been changed to test for a return code from DB2 Bind Manager. Otherwise, bind the new DBRM and copy it to the DBRMLIB.

Module: BNDAVB

BND023S **DB2 BIND MANAGER PROCESSING TERMINATED DUE TO UNRECOVERABLE ERROR**

Explanation: DB2 Bind Manager has encountered a severe error and cannot continue. This message is normally preceded by another message stating the specific error or cause of failure.

Programmer response: Save all output. Gather the following materials and contact IBM Software Support:

- the preexisting (old) DBRM
- the new DBRM, if one was created
- all SYSOUT from the job that failed (the entire job, not just the Bind Manager job step)

Module: BNDAVB

BND024S **** ERROR ** SSID DD STATEMENT IS PRESENT BUT NO SUBSYSTEM WAS SPECIFIED**

Explanation: DBRM Generation was requested by including an SSID DD statement in the JCL, but no subsystem name was specified. Therefore, DB2 Bind Manager did not know which DB2 system catalog to use for generating the DBRM. The correct syntax is: //SSIDxyz DD DUMMY . . . where *xyz* is the 1- to 4-character subsystem ID. In this example, DB2 Bind Manager is requested to use subsystem DSN1 to generate the DBRM: //SSIDDSN1 DD DUMMY.

Programmer response: Add the required DD statement to the JCL.

Module: BNDAVB

BND025I **UNABLE TO OPEN SYSPRINT
UNABLE TO EXTRACT DBRM FROM
CATALOG SYMPTOMS=
xxxxxxxx/yyyyyyyy**

Explanation: DBRM Generation was requested but DB2 Bind Manager was unable to extract the relevant information from the catalog. This error normally occurs when the program is being compiled for the first time because the catalog does not contain an existing

copy of the DBRM. A bind will be required.

Programmer response: This is condition is usually not an error. Therefore, a response is not required. If you are certain that the catalog contains a valid DBRM for the program that is being compiled, save all output and contact IBM Software Support.

Module: BNDAVB1

BND026I THE "OLD" DBRM WAS FOUND IN PLAN *planname* IN THE DB2 CATALOG

Explanation: DBRM Generation was performed. The DBRM was reconstructed based on the DBRM in plan *planname*.

Programmer response: None.

Module: BNDAVB

BND027I THE "OLD" DBRM WAS FOUND IN VERSION *version* IN THE DB2 CATALOG

Explanation: DBRM Generation was performed. The DBRM was reconstructed based on the package in version *version*.

Programmer response: None.

Module: BNDAVB

BND030W BNDWK1 DD STATEMENT MISSING, DBRM NOT GENERATED

Explanation: The DD statement is missing.

Programmer response: Add a BNDWK1 DD statement to the JCL and resubmit the job.

Module: BNDAVB

BND031E NEW DBRM IS NOT VALID

Explanation: The DBRM produced by this precompile is damaged or corrupted. Specifically, the DBRM eyecatcher text is missing where it should be present.

Programmer response: Contact IBM Software Support.

Module: BNDAVB

BND098I BIND MANAGER PROCESSING TERMINATED, !! BIND REQUIRED !!

Explanation: The process ended when it determined that a DB2 bind is required. This message is normally accompanied by return code 4. If this message is preceded by message BND014E, the words **BIND REQUIRED** in the message text are replaced with **CORRECT PRECOMPILER ERRORS**.

Programmer response: None, if the variable text is **BIND REQUIRED** and the bind procedure has been changed to test for a return code from DB2 Bind

Manager. Otherwise, depending on the message text, fix the source program and resubmit the job or bind the new DBRM and copy it to the DBRMLIB.

Module: BNDAVB

BND099S BIND MANAGER PROCESSING FAILED DURING DBRM UPDATE, DELETE THE DBRM AND RERUN

Explanation: DB2 Bind Manager has failed during the DBRM update. This message should be accompanied by a U999 ABEND and SYSUDUMP. The DBRM may have been corrupted.

Programmer response: Ensure that DBMRLIB specifies the correct library and that the user who submitted the job has authority to update the library. If the problem persists, contact IBM Software Support.

Module: BNDAVB

BND200S SYSPRINT DD STATEMENT MISSING

Explanation: The JCL does not contain a SYSPRINT DD statement.

Programmer response: Add a SYSPRINT DD statement and resubmit the job.

Module: BNDAVB1

BND201S UNABLE TO OPEN SYSPRINT

Explanation: SYSPRINT could not be opened for output.

Programmer response: Ensure that a valid SYSPRINT DD statement is present in the JCL and that you have update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND202S SYSUT1 DD STATEMENT MISSING

Explanation: The JCL does not contain a SYSUT1 DD statement.

Programmer response: Add a SYSUT1 DD statement and resubmit the job.

Module: BNDAVB1

BND203S UNABLE TO OPEN SYSUT1

Explanation: SYSUT1 could not be opened for output.

Programmer response: Ensure that a valid SYSUT1 DD statement is present in the JCL and that you have update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND206S OLDDBRM DD STATEMENT MISSING

Explanation: The JCL does not contain an OLDDBRM DD statement.

Programmer response: Add an OLDDBRM DD statement and resubmit the job.

Module: BNDAVB1

BND207S UNABLE TO OPEN OLDDBRM

Explanation: OLDDBRM could not be opened for input.

Programmer response: Ensure that the JCL contains an OLDDBRM DD statement, that the statement references a valid DBRM, and that the user has read access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND208S OLDDBRM — INCORRECT DCB ATTRIBUTES (LRECL)

Explanation: The data set that was referenced by OLDDBRM has an incorrect format.

Programmer response: Ensure that the JCL contains an OLDDBRM DD statement and that the statement references a valid DBRM library with LRECL=80. Resubmit the job.

Module: BNDAVB1

BND209S OLDDBRM — INCORRECT DCB ATTRIBUTES (RECFM)

Explanation: The data set that was referenced by OLDDBRM has an incorrect format.

Programmer response: Ensure that the JCL contains an OLDDBRM DD statement and that the statement references a valid DBRM library with RECFM=FB. Resubmit the job.

Module: BNDAVB1

BND210S NEWDBRM DD STATEMENT MISSING

Explanation: The JCL does not contain a NEWDBRM DD statement.

Programmer response: Add an OLDDBRM DD statement and resubmit the job.

Module: BNDAVB1

BND211S UNABLE TO OPEN NEWDBRM

Explanation: NEWDBRM could not be opened for input.

Programmer response: Ensure that the JCL contains an NEWDBRM DD statement, that the statement references a valid DBRM, and that the user has read access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND212S NEWDBRM — INCORRECT DCB ATTRIBUTES (LRECL)

Explanation: The data set that was referenced by NEWDBRM has an incorrect format.

Programmer response: Ensure that the JCL contains a NEWDBRM DD statement and that the statement references a valid DBRM library with LRECL=80. Resubmit the job.

Module: BNDAVB1

BND213S NEWDBRM — INCORRECT DCB ATTRIBUTES (RECFM)

Explanation: The data set that was referenced by NEWDBRM has an incorrect format.

Programmer response: Ensure that the JCL contains a NEWDBRM DD statement and that the statement references a valid DBRM library with RECFM=FB. Resubmit the job.

Module: BNDAVB1

BND214S OBJECT DD STATEMENT MISSING

Explanation: The JCL does not contain an OBJECT DD statement.

Programmer response: Add an OBJECT DD statement and resubmit the job.

Module: BNDAVB1

BND215S UNABLE TO OPEN OBJECT

Explanation: OBJECT could not be opened successfully.

Programmer response: Ensure that the JCL contains an OBJECT DD statement, that the statement references a data set or PDS member, and that the user has update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

**BND216S OBJECT — INCORRECT DCB
ATTRIBUTES (LRECL)**

Explanation: The data set that was referenced by OBJECT has an incorrect format.

Programmer response: Ensure that the JCL contains an OBJECT DD statement and that the statement references a valid data set or PDS member with LRECL=80. Resubmit the job.

Module: BNDAVB1

**BND217S OBJECT — INCORRECT DCB
ATTRIBUTES (RECFM) E**

Explanation: The data set that was referenced by OBJECT has an incorrect format.

Programmer response: Ensure that the JCL contains an OBJECT DD statement and that the statement references a valid data set or PDS member with RECFM=FB. Resubmit the job.

Module: BNDAVB1

**BND218I BNDAVB1 TERMINATED DUE TO
ERROR**

Explanation: BNDAVB1 has ended with one or more errors.

Programmer response: None. This message is normally preceded by one or more error messages that describe the specific errors.

Module: BNDAVB1

**BND219E DATA REFERENCED BY THE
OLDDBRM DD IS NOT A VALID
DBRM**

Explanation: The DBRM that was referenced by the OLDDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the OLDDBRM DD statement so that it references a valid DBRM and resubmit the job.

Module: BNDAVB1

**BND220E DATA REFERENCED BY THE
NEWDBRM DD IS NOT A VALID
DBRM**

Explanation: The DBRM that was referenced by the NEWDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the NEWDBRM DD statement so that it references a valid DBRM and resubmit the job.

Module: BNDAVB1

**BND221E TIME STAMP HAS INVALID FORMAT,
"LEVEL" PRECOMPILER OPTION NOT
SUPPORTED**

Explanation: The old DBRM was generated by using the DB2 precompiler/coprocessor LEVEL option. This option is not supported by DB2 Bind Manager.

Programmer response: No action is required. However, DB2 Bind Manager will set a return code 4 to require a bind. To avoid future binds, run without the LEVEL option.

Module: BNDAVB1

**BND222E TIME STAMP HAS INVALID FORMAT,
"LEVEL" PRECOMPILER OPTION NOT
SUPPORTED**

Explanation: The new DBRM was generated by using the DB2 precompiler/coprocessor LEVEL option. This option is not supported by DB2 Bind Manager.

Programmer response: No action is required. However, DB2 Bind Manager will set a return code 4 to require a bind. To avoid future binds, resubmit the job without the LEVEL option.

Module: BNDAVB1

**BND223E DATA REFERENCED BY THE
OLDDBRM DD IS NOT A VALID
DBRM**

Explanation: The DBRM that was referenced by the OLDDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the OLDDBRM DD statement so that it references a valid DBRM and resubmit the job.

Module: BNDAVB1

**BND224E DATA REFERENCED BY THE
NEWDBRM DD IS NOT A VALID
DBRM**

Explanation: The DBRM that was referenced by the NEWDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the NEWDBRM DD statement so that it references a valid DBRM and resubmit the job.

Module: BNDAVB1

**BND225I DB2 VERSION/RELEASE MISMATCH
BETWEEN OLD/NEW DBRMS**

Explanation: The old and new DBRMs were generated by different releases of the DB2 precompiler. A bind will be required.

BND226E • BND233E

Programmer response: None.

Module: BNDAVB1

BND226E DB2 VERSION/RELEASE MISMATCH BETWEEN OLD/NEW DBRMS

Explanation: The old and new DBRMs were created under different releases of DB2.

Programmer response: An initial bind must be done under the new release; afterwards DB2 Bind Manager will function normally with the new DBRM format. To allow for possible modifications to the optimizer, it is always best to bind existing applications when the DB2 version or release level changes.

Module: BND1AVB

BND226I DBRM COMPARE FAILED — NON-MATCHING CHARACTER SETS (CCSIDS)

Explanation: The old and new DBRMs were generated with different character sets. A bind will be required. This error can occur when the old DBRM is in EBCDIC and the new DBRM is in unicode.

Programmer response: None.

Module: BNDAVB1

BND227E DATA REFERENCED BY THE OLDDBRM DD IS NOT A VALID DBRM

Explanation: The DBRM that was referenced by the OLDDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the OLDDBRM DD statement so that it references a valid DBRM.

Module: BNDAVB1

BND228E DATA REFERENCED BY THE NEWDBRM DD IS NOT A VALID DBRM

Explanation: The DBRM that was referenced by the NEWDBRM DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the NEWDBRM DD statement so that it references a valid DBRM.

Module: BNDAVB1

BND229E DELETION OF SOURCE STATEMENTS HAS INVALIDATED THE DBRM - ADD COMMENT LINES TO CORRECT

Explanation: This error occurs when deleting statements from a very large source module. Because of a limitation of the DB2 precompiler, the internal format

of the DBRM for a COBOL program changes when the growth of a source program forces at least one SQL statement to have a relative line number greater than 9999. This condition does not pose a problem when it occurs, but it does cause a downward compatibility problem later if the lines are subsequently deleted, causing SQL statements to again have line numbers that are less than 10,000. This scenario creates a condition that DB2 Bind Manager cannot handle, so the default is to require a bind.

Programmer response: If the SQL structure has not been modified, you can circumvent this situation by using the following procedure:

1. Restore the old DBRM (the version that was saved in the job step that precedes the precompiler step).
2. Rather than deleting source statements, comment them out. This will leave the relative line numbers unchanged.
3. Rerun the compile job.

Module: BNDAVB1

BND230I BIND NOT REQUIRED

Explanation: Bind Manager has returned a condition code 0 to indicate that no bind is needed.

Programmer response: None.

Module: BNDAVB1

BND232E OBJECT CODE SCAN FAILED — CONSISTENCY TOKEN NOT FOUND

Explanation: A DB2 timestamp (consistency token) was not found in the generated object code.

Programmer response: If there are no SQL statements in the program that was being compiled, this condition is normal and requires no response. If the program contains static SQL, this condition reflects a probable program logic error in Bind Manager. In the latter case, save all output and contact IBM Software Support.

Module: BNDAVB1

BND233E OBJECT CODE SCAN FAILED — CONSISTENCY TOKEN NOT FOUND

Explanation: The generated object code does not contain a DB2 timestamp (consistency token).

Programmer response: If there are no SQL statements in the program that was being compiled, this condition is normal and requires no response. If the program contains static SQL, this condition reflects a probable program logic error in Bind Manager. In the latter case, save all output and contact IBM Software Support.

Module: BNDAVB1

**BND234E OBJECT CODE SCAN FAILED —
CONSISTENCY TOKEN NOT FOUND**

Explanation: A DB2 timestamp (consistency token) was not found in the generated object code.

Programmer response: If there are no SQL statements in the program that was being compiled, then this condition is normal and requires no response. If the program contains static SQL, then this condition reflects a probable program logic error in DB2 Bind Manager. In the latter case, save all output and contact IBM Software Support.

Module: BNDAVB1

BND235S Premature end of file -- old DBRM

Explanation: The DBRM that was referenced by OLDDBRM is damaged or truncated.

Programmer response: Correct the OLDDBRM DD statement so that it references a valid DBRM and resubmit the job.

Module: BNDAVB1

BND236S UNABLE TO OPEN OLDDBRM

Explanation: OLDDBRM could not be opened for input.

Programmer response: Ensure that the JCL contains an OLDDBRM DD statement, that the statement references a valid DBRM, and that the user has read access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND237S UNABLE TO OPEN NEWDBRM

Explanation: NEWDBRM could not be opened for input.

Programmer response: Ensure that the JCL contains an NEWDBRM DD statement, that the statement references a valid DBRM, and that the user has read access to the indicated data set. Resubmit the job.

Module: BNDAVB1

BND238S UNABLE TO OPEN *ddname* OBJECT

Explanation: OBJECT could not be opened successfully.

Programmer response: Ensure that the JCL contains an OLDDBRM DD statement, that the statement references a data set or PDS member, and that the user has update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

**BND239S TOO MANY CSECTS IN OBJECT
MODULE (LIMIT EXCEEDED)**

Explanation: The object program that was being processed contains more CSECTS than will fit in an internal work area. This condition should not occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

**BND240E MODULE CONSTRUCTION ERROR
(NO CSECTS)**

Explanation: The object program that was specified by the OBJECT DD statement is empty, truncated, or has an incorrect format.

Programmer response: Correct the OBJECT DD statement so that it references a data set or PDS member that contains a valid object deck and resubmit the job.

Module: BNDAVB1

BND241S UNABLE TO OPEN *ddname* OBJECT

Explanation: OBJECT could not be opened successfully.

Programmer response: Ensure that the JCL contains an OBJECT DD statement, that the statement references a data set or PDS member, and that the user has update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

**BND242S INTERNAL PROCESSING ERROR
(INDEX ENTRY NOT FOUND)**

Explanation: A referential integrity mismatch exists between two internal work tables used by the program. This condition should not occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

BND243S UNABLE TO OPEN *ddname* OBJECT

Explanation: OBJECT could not be opened successfully.

Programmer response: Ensure that the JCL contains an OBJECT DD statement, that the statement references a data set or PDS member, and that the user has update access to the indicated data set. Resubmit the job.

Module: BNDAVB1

**BND244S INTERNAL PROCESSING ERROR
(INDEX ENTRY NOT FOUND)**

Explanation: A referential integrity mismatch exists between two internal work tables that are used by the program. This condition should not occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

**BND245S INTERNAL PROCESSING ERROR
(INDEX ENTRY NOT FOUND)**

Explanation: A referential integrity mismatch exists between two internal work tables that are used by the program. This condition should not occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

**BND246S INTERNAL PROCESSING ERROR
(UNABLE TO DETERMINE CORRECT
STATEMENT NUMBER)**

Explanation: A referential integrity mismatch exists between two internal work tables used by the program. This condition should not occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

BND248S MORE THAN 3000 SQL STATEMENTS

Explanation: DB2 Bind Manager has detected a DBRM that contains more than 3,000 SQL calls. This exceeds the maximum supported program size.

Programmer response: Restructure the application to reduce the number of SQL calls and resubmit the job.

Module: BNDAVB1

**BND249S INTERNAL PROCESSING ERROR
(UNABLE TO DETERMINE CORRECT
STATEMENT NUMBER) RECOMPILE
PROGRAM WITHOUT BNDAVB1 (A
BIND WILL BE REQUIRED)**

Explanation: The object code is in a format that is not recognized by BNDAVB1. This condition applies only to PL/I programs and should never occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDAVB1

**BND301E DBRM ALLOCATION FAILED,
RC=*return code***

Explanation: DBRM Checker was unable to allocate the specified DBRMLIB.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library.

Module: BNDC002

**BND302E DBRM ALLOCATION FAILED,
RC=*return code***

Explanation: DBRM Checker was unable to allocate the specified DBRMLIB.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDC002

**BND303E UNABLE TO ALLOCATE LOADLIB,'
RC=*return code***

Explanation: DBRM Checker was unable to allocate the specified load library.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing load library.

Module: BNDC002

**BND306E MEMBER *dbrmname* NOT FOUND IN
*dbrmlib***

Explanation: The specified DBRM was not found in the specified library.

Programmer response: Specify the name of a valid (existing) DBRM.

Module: BNDC002

**BND307E LISTDSI FAILED FOR
DBRMLIB—*reason***

Explanation: The specified DBRMLIB could not be accessed. *Reason* is a description of the specific error encountered.

Programmer response: Determine why the data set was unavailable (based on reason in the message), and take appropriate corrective action.

Module: BNDC002

BND308E *dbrmlib* IS NOT A VALID DBRM LIBRARY

Explanation: The specified DBRMLIB is not valid. The DBRMLIB must be a partitioned data set (PDS) with the attributes RECFM=FB and LRECL=80. One or more of these attributes is not present.

Programmer response: Specify the name of a valid DBRM library.

Module: BNDC002

BND309E LISTDSI FAILED FOR LOADLIB—*reason code*

Explanation: The specified load library could not be accessed.

Programmer response: Determine why the data set was unavailable (based on reason in the message), and take appropriate corrective action.

Module: BNDC002

BND310E *dbrmlib* IS NOT A VALID DBRM LIBRARY

Explanation: The specified load library is not valid. The load library must be a partitioned data set (PDS) with RECFM=U. One or more of these attributes is not correct.

Programmer response: Specify the name of a valid load library.

Module: BNDC002

BND311E DBRMLIB DATA SET NOT CATALOGED

Explanation: The DBRMLIB data set could not be found.

Programmer response: Specify the name of an existing DBRM library.

Module: BNDC002

BND312E LOADLIB DATA SET NOT CATALOGED

Explanation: The LOADLIB data set could not be found.

Programmer response: Specify the name of an existing load library.

Module: BNDC002

BND313E LMINIT FAILED, RC=*return code*

Explanation: A failure has occurred in the ISPF interface.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDC002

BND314E LMOPEN FAILED, RC=*return code*

Explanation: A failure has occurred in the ISPF interface.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDC002

BND315E A DBRM NAME MUST BE SPECIFIED OR SELECTED FROM THE MEMBER LIST

Explanation: A DBRM name was not supplied.

Programmer response: Specify the name of an existing DBRM.

Module: BNDC002

BND316E LMINIT FAILED, RC=*return code*

Explanation: A failure has occurred in the ISPF interface.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDC002

BND317E LMOPEN FAILED, RC=*return code*

Explanation: A failure has occurred in the ISPF interface.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDC002

BND318E A LOAD MODULE NAME MUST BE SPECIFIED OR SELECTED FROM THE MEMBER LIST

Explanation: A load module name was not supplied.

BND319E • BND354E

Programmer response: Specify the name of an existing load module.

Module: BNDC002

BND319E MEMBER *dbrmname* NOT FOUND IN *loadlib*

Explanation: The specified load module was not found in the specified library.

Programmer response: Specify the name of a valid (existing) load module.

Module: BNDC002

BND320E SELECT ONE OF THE LISTED OPTIONS

Explanation: A valid menu option was not specified.

Programmer response: Select an existing option from the menu.

Module: BNDC002

BND321S LOAD MODULE BNDA005 NOT FOUND

Explanation: Load module BNDA005 cannot be located. An error probably occurred during installation. Either the correct STEPLIB was not specified, or the module was not copied to the library that was being used.

Programmer response: Correct the installation error and retry.

Module: BNDC002

BND322S DBRMC NEEDS TO BE CUSTOMIZED FOR THIS INSTALLATION

Explanation: An error probably occurred during installation. The DBRMC CLIST needs to be modified before the ISPF dialog can be used. The CLIST contains comments that describe the required modifications.

Programmer response: Customize the CLIST appropriately and retry.

Module: BNDC002

BND323E UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, RC=*return code*

Explanation: DB2 Bind Manager could not establish a connection to the requested DB2 subsystem.

Programmer response: Verify that the SSID that you specified is correct and that the subsystem is running and available.

Module: BNDC002

BND324E PREPARE FAILED, SQLCODE *sqlcode*

Explanation: An unexpected DB2 error occurred. In the message text, *sqlcode* is diagnostic information.

Programmer response: Contact IBM Software Support.

Module: BNDC002

BND325E OPEN CURSOR FAILED, SQLCODE *sqlcode*

Explanation: An unexpected DB2 error occurred. In the message text, *sqlcode* is diagnostic information.

Programmer response: Contact IBM Software Support.

Module: BNDC002

BND326I SUBSYSTEM *remotesubsystem* IS NOT KNOWN TO *localsubsystem*

Explanation: DB2 Bind Manager was unable to connect to the remote DB2 subsystem.

Programmer response: Verify that the subsystem names are correct and that the requisite DDF connections are in place and working.

Module: BNDC002

BND327E FETCH C1 FAILED, SQLCODE *sqlcode*

Explanation: An unexpected DB2 error occurred.

Programmer response: Contact IBM Software Support.

Module: BNDC002

BND353E IEWBIND STARTED FAILED, REASON CODE *reason code*

Explanation: A failure has occurred in the Binder API.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDA005

BND354E IEWBIND ENDD FAILED, REASON CODE *reason code*

Explanation: A failure has occurred in the Binder API.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDA005

BND355E **IEWBIND CREATEW FAILED,
REASON CODE** *reason code*

Explanation: A failure has occurred in the Binder API.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDA005

BND356S **INTERNAL PROCESSING ERROR,
TYPE 6, REASON CODE** *reason code*

Explanation: A severe error has occurred.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDA005

BND357S **ERROR - UNABLE TO ACCESS DBRM
LIBRARY**

Explanation: DBRM Checker was unable to OPEN the DBRM library.

Programmer response: Check the job log for messages indicating the cause of the failure. Take appropriate corrective action and retry.

Module: BNDA005

BND358S **ERROR - PREMATURE END OF FILE
READING DBRM**

Explanation: The DBRM ended prematurely. The DBRM library or the individual member might be corrupted.

Programmer response: Make sure the DBRM library and member have not been overwritten or otherwise damaged. If the data appears valid, Contact IBM Software Support.

Module: BNDA005

BND359E **ERROR - LOAD MODULE NOT
FOUND**

Explanation: The load module was not found in the designated load library.

Programmer response: Specify the name of a valid (existing) load module and retry.

Module: BNDA005

BND360E **IEWBIND CREATEW FAILED,
REASON CODE** *reason code*

Explanation: A failure has occurred in the Binder API.

Programmer response: Ensure that the data set name that was entered on the panel refers to a valid, existing DBRM library. If this does not resolve the problem, contact IBM Software Support.

Module: BNDA005

BND361E **DBRM FORMAT ERROR, REASON
CODE 11**

Explanation: The DBRM contains unrecognizable or invalid data. The DBRM library or the individual member might be corrupted.

Programmer response: Make sure the DBRM library and member have not been overwritten or otherwise damaged. If the data appears valid, contact IBM Software Support.

Module: BNDA005

BND362E **DBRM FORMAT ERROR, REASON
CODE 12**

Explanation: The DBRM contains unrecognizable or invalid data. The DBRM library or the individual member might be corrupted.

Programmer response: Make sure the DBRM library and member have not been overwritten or otherwise damaged. If the data appears valid, contact IBM Software Support.

Module: BNDA005

BND364E **UNABLE TO CONNECT TO
SUBSYSTEM** *subsystem*

Explanation: DBRM Checker is unable to establish a connection to DB2 subsystem *subsystem*.

Programmer response: Ensure that a valid subsystem name has been specified and that the subsystem is running and available.

Module: BNDA005

BND365S **LOAD FAILED FOR MODULE
BNDA009**

Explanation: DBRM Checker is unable to access one of its component programs.

Programmer response: Probable installation error. Make sure that program BNDA009 is available to the TSO session, either through a STEPLIB or through the system link list.

Module: BNDA005

BND366E *dbrmname* NOT FOUND IN *subsystem*,
SQLCODE *sqlcode*

Explanation: No entry corresponding to the target DBRM was found in the DB2 catalog.

Programmer response: Ensure that the DBRM specified exists (has been bound) in the specified DB2 subsystem. If the DBRM and subsystem are valid, contact IBM Software Support.

Module: BNDA005

BND367E UNABLE TO ACCESS PLAN BNDA009

Explanation: Program BNDA009 has not been successfully bound in the DB2 subsystem that you are accessing.

Programmer response: Probable installation error. Make sure that a successful Bind is run for program BNDA009 and retry.

Module: BNDA005

BND368E DB2 INTERFACE ERROR (NONZERO
RETURN CODE FROM BNDA009)

Explanation: DBRM Checker encountered an unexpected DB2 error.

Programmer response: Ensure that a valid subsystem name has been specified and that the subsystem is running and available.

Module: BNDA005

BND369I WARNING: MULTIPLE INSTANCES
OF *dbrmname* FOUND IN *subsystem*

Explanation: Multiple instances of DBRM *dbrmname* were found in subsystem *subsystem*, but none have time stamps that agree with the input DBRM.

Programmer response: None.

Module: BNDA005

BND374S THE BIND FOR PROGRAM BNDA009
IS NOT CURRENT

Explanation: The version of program BNDA009 that was being run does not match the last one that was bound in the DB2 subsystem; DBRM Checker cannot run successfully.

Programmer response: Probable installation error. Make sure the version of DBRM Checker that was being run was successfully bound in the subsystem that is being used.

Module: BNDA005

BND375I WARNING: MULTIPLE INSTANCES
OF *dbrmname* FOUND IN *subsystem*

Explanation: Multiple instances of DBRM *dbrmname* were found in subsystem *subsystem*. At least one has a timestamp that matches the DBRM.

Programmer response: None.

Module: BNDA005

BND376S ERROR — WORK AREA OVERFLOW
— TOO MANY INSTANCES OF DBRM
IN SUBSYSTEM, UNABLE TO
PROCESS

Explanation: DBRM Check uses an internal work area that can hold up to 182 catalog entries that match the input DBRM. More than this number were found.

Programmer response: Verify that the correct DBRM is being inquired against and that the correct subsystem is being referenced. If the problem persists, contact IBM Software Support.

Module: BNDA005

BND377S UNABLE TO FETCH MODULE
ISPLINK

Explanation: DBRM Checker cannot access the ISPLINK API routine.

Programmer response: Make sure that the ISPLINK routine is available to your TSO session, either through STEPLIB or the system link list.

Module: BNDA005

BND378S UNABLE TO FETCH MODULE
ISPEXEC

Explanation: DBRM Checker cannot access the ISPEXEC API routine.

Programmer response: Make sure that the ISPEXEC routine is available to your TSO session, either through STEPLIB or the system link list.

Module: BNDA005

BND382W IEWBIND CREATEW RC=4, REASON
CODE *reason code*

Explanation: The MVS loader returned a condition code 4 with reason code *reason code* when bringing a load module into memory for examination. This can occur when the binder detects unusual or anomalous data in the load module, for example, a 2 byte address constant. This condition might indicate the possibility of problems when running the load module. However, there is no direct effect on DB2 Bind Manager processing; the timestamp scan proceeds normally.

Programmer response: None.

Module: BNDA005

BND401S DB2 API not available, symptom=*nnn*

Explanation: DB2 Bind Manager failed to establish a connection to the REXX DB2 API. Either the API is not installed properly, or it resides in a library not available to your TSO session. In the message text, *nnn* is the return code received when DB2 Bind Manager attempted to connect to the API.

Programmer response: Ensure that the REXX DB2 API is installed on your system and that it can be accessed by your TSO session.

Module: BND005

BND402E Enter one of the listed options

Explanation: The value entered is not one of the valid choices for the current menu.

Programmer response: Enter one of the valid choices and retry the request.

Module: BND005

BND404S PREPARE *sss* failed, SQLCODE=*nnn*

Explanation: DB2 Bind Manager was unable to PREPARE a dynamic SQL statement. In the message text, *sss* is an internal diagnostic code indicating which statement was being prepared when the failure occurred, and *nnn* is the SQLCODE returned by DB2.

Programmer response: Analyze the SQLCODE to determine why the PREPARE was unsuccessful. SQLCODES are described in the Codes manual for the Version of DB2 installed in your subsystem. Correct the condition causing the failure and retry the request.

Module: BND005

BND405S OPEN *sss* failed, SQLCODE=*nnn*

Explanation: DB2 Bind Manager was unable to OPEN a cursor. In the message text, *sss* is an internal diagnostic code indicating which cursor was being opened when the failure occurred, and *nnn* is the SQLCODE returned by DB2.

Programmer response: Analyze the SQLCODE to determine why the OPEN was unsuccessful. SQLCODES are described in the *Codes* manual for the version of DB2 installed in your subsystem. Correct the condition causing the failure and try the request.

Module: BND005

BND406S TBCREATE failed, *ttt*, symptom=*nnn*

Explanation: DB2 Bind Manager was unable to create an ISPF table to displaying the results of a request. In the message text, *ttt* is the name of the table for which the create operation failed, and *nnn* is the return code issued by the TBCREATE service of ISPF.

Programmer response: Analyze the return code to determine why the table could not be created. The return codes for ISPF services are described in the *ISPF Services Guide* for the version and release of ISPF in use on your system. Correct the condition causing the failure and retry the request.

Module: BND005

BND408E Only one entry can be processed at a time

Explanation: Line commands were entered next to more than one of the displayed results in a list. Only one line command at a time is valid.

Programmer response: Enter a single valid line command. When that command is processed, enter the next one, and so forth, until all of the desired line commands have been processed individually.

Module: BND005

BND409S FETCH *ccc* failed, SQLCODE=*nnn*

Explanation: DB2 Bind Manager was unable to execute a FETCH for an open cursor. In the message text, *ccc* is an internal diagnostic code indicating which cursor was being fetched when the failure occurred, and *nnn* is the SQLCODE returned by DB2.

Programmer response: Analyze the SQLCODE to determine why the FETCH was unsuccessful. SQLCODES are described in the *Codes* manual for the version of DB2 installed in your subsystem. Correct the condition causing the failure and rerun.

Module: BND005

BND411E *dsn* could not be allocated. Reason: *reason_text*

Explanation: The specified DBRM library could not be accessed. The name might have been misspelled, the library might not exist, or it might be inaccessible for some other reason. In the message text, *dsn* is the data set name of the DBRM library and *reason_text* is a brief description of why it is unavailable.

Programmer response: Correct the condition that is preventing the DBRM library from being accessed and retry the request.

Module: BND005

BND412E Invalid DBRM library – *reason_text*

Explanation: The specified data set is not a valid DBRM library. Either it is not partitioned (not a PDS), or it has an invalid record format (RECFM) or an incorrect record length (LRECL). In the message text, *reason_text* is a short phrase identifying the invalid attribute.

Programmer response: Specify the name of a valid DBRM library and retry the request.

Module: BND005

BND413E Unable to allocate DBRMLIB,
reason=*nnn*

Explanation: The specified DBRM library exists but could not be allocated for output. The library might be in use by another user or might be unavailable for some other reason. In the message text, *nnn* is the return code from the TSO ALLOCATE command.

Programmer response: Analyze the return code to determine why the DBRM library is inaccessible. Return codes for the ALLOCATE command are described in the *TSO/E Command Reference* manual for the release of z/OS installed on your system. Correct the condition preventing the allocation and retry the request.

Module: BND005

BND414S Unable to connect to subsystem *sss*,
reason=*nnn*

Explanation: DB2 Bind Manager was unable to establish a connection to the specified DB2 subsystem. The subsystem name might have been misspelled, the subsystem might not exist or might not be running, or some other problem can be preventing a successful connection. In the message text, *sss* is the name of the subsystem being accessed and *nnn* is the return code issued by the CONNECT request.

Programmer response: Analyze the return code to determine why the connection failed. The return codes for the CONNECT service of the Call Attachment Facility are documented in the *Application Programming and SQL Guide* for the version of DB2 installed in your subsystem. Correct the condition causing the failure and retry the request.

Module: BND005

BND415E '*ccc*' is not a valid line command; 'S' and '/' are the only acceptable entries

Explanation: An invalid line command was entered next to one of the displayed entries. S and the forward slash / are the only valid line commands for this display. In the message text, *ccc* is the invalid line command that was specified.

Programmer response: Retry the request with one of the valid line commands.

Module: BND005

BND416E Unexpected value *nnn* found in column *ccc* of *ttt*; DBRM not generated

Explanation: DB2 Bind Manager was unable to reconstruct the DBRM because the catalog table contained invalid data. In the message text, *nnn* is the invalid value, *ccc* is the name of the column containing it, and *ttt* is the name of the catalog table where it was encountered.

Programmer response: None. It is not possible to reconstruct this DBRM from available data.

Module: BND005.

**BND500S INDEX UTILITY FAILED DURING
INITIALIZATION,
SYMPTOMS=*symptoms***

Explanation: The Index Utility program experienced an environmental failure or logic error and was unable to successfully initialize.

Programmer response: Probable installation or configuration error. Verify that DB2 Bind Manager was installed correctly.

Module: BND006

**BND502S DBRM CHECKER FAILED TO UPDATE
VARIABLES DURING TERMINATION,
SYMPTOM=*symptom***

Explanation: A failure occurred in the ISPF interface.

Programmer response: Verify that the correct profile data set is allocated to ddname ISPPROF, that the data set is not full or in need of compression, and resubmit the job.

Module: BND006

**BND503E UNABLE TO CONNECT TO
SUBSYSTEM *subsystem***

Explanation: DBRM Checker was unable to establish a connection to DB2 subsystem *subsystem*.

Programmer response: Verify that the SSID that you specified is correct and that the subsystem is running and available.

Module: BND006

**BND505S UNABLE TO DISCONNECT FROM
SUBSYSTEM *subsystem***

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem

name, ensure that the subsystem is running and available, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BND006

BND506E *command IS NOT A VALID COMMAND*

Explanation: An unrecognized or otherwise invalid command was entered in the command field.

Programmer response: Enter a valid command or leave the command field blank.

Module: BND006

BND507S **DECLARE CURSOR FAILED,**
*SYMPATOM=**symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND508S **PREPARE FAILED, SYMPTOM=***symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND509S **TBCREATE FAILED,**
*SYMPTOM=**symptom*

Explanation: The ISPF API unexpectedly returned a non-zero condition code.

Programmer response: Verify that the correct profile data set is allocated to ddname ISPPROF and that the data set is not full or in need of compression. If the error persists, gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND510S **OPEN CURSOR FAILED,**
*SYMPTOM=**symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND511S **TBADD FAILED FOR PLAN ENTRY,**
*SYMPTOM=**symptom*

Explanation: The ISPF API unexpectedly returned a non-zero condition code.

Programmer response: Verify that the correct profile data set is allocated to ddname ISPPROF and that the data set is not full or in need of compression. If the error persists, gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND512S **SQL FETCH FAILED,**
*SYMPTOM=**symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND513I **NO INDEXES ARE DEFINED FOR** *table*

Explanation: The table you entered has no indexes.

Programmer response: None.

Module: BND006

BND514E **MORE THAN ONE ITEM WAS
SELECTED (ONLY ONE ITEM AT A
TIME CAN BE PROCESSED)**

Explanation: You selected more than one item from the displayed list.

Programmer response: Select only one item at a time for processing.

Module: BND006

BND515E **INVALID SELECTION - THE ONLY
VALID SELECTION CHARACTERS
ARE '/', 'S', AND 'C'**

Explanation: The selection character entered was invalid.

Programmer response: Enter one of the valid characters.

Module: BND006

BND516S **PREPARE FAILED, SYMPTOM=***symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND517S **TBCREATE FAILED,**
SYMPTOM=*symptom*

Explanation: The ISPF API unexpectedly returned a non-zero condition code.

Programmer response: Verify that the correct profile data set is allocated to ddname ISPPROF and that the data set is not full or in need of compression. If the error persists, gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND518S **OPEN CURSOR FAILED,**
SYMPTOM=*symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND519S **TBADD FAILED, SYMPTOM=*symptom***

Explanation: The ISPF API unexpectedly returned a non-zero condition code.

Programmer response: Verify that the correct profile data set is allocated to ddname ISPPROF and that the data set is not full or in need of compression. If the error persists, gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND520S **SQL FETCH FAILED -**
SYMPTOM=*symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND522S **LOGIC ERROR — SYMPTOMS=BND/
ISPMSG/*nnn/mmm***

Explanation: A program logic error was detected. Internal trace codes are provided for diagnostic purposes. This condition should not occur in normal processing.

Programmer response: Gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND523S **PREPARE FAILED, SYMPTOM=*symptom***

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND524S **OPEN CURSOR FAILED,**
SYMPTOM=*symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND525S **SQL FETCH FAILED,**
SYMPTOM=*symptom*

Explanation: An unexpected DB2 error occurred.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is running and available, and resubmit the job.

Module: BND006

BND526E **BAD SYNTAX IN TABLE NAME**

Explanation: An invalid table name was entered.

Programmer response: Enter a valid table name.

Module: BND006

BND527S **DBRM CHECKER INDEX UTILITY -
LOGIC ERROR - SYMPTOM=*symptom***

Explanation: DB2 Bind Manager module BND006 attempted to display an undefined message number. This condition should not occur in normal processing.

Programmer response: Gather the ISPF log data set and a screen capture of the message and contact IBM Software Support.

Module: BND006

BND528E **INVALID QUALIFIER -- MUST BE 1
TO 8 ALPHANUMERIC
CHARACTERS, THE FIRST
ALPHABETIC**

Explanation: The syntax for the high-level qualifier is invalid. Qualifiers must be from 1 to 8 characters in length. Valid characters include the letters A through Z, the digits 0 through 9, and the characters \$, @, and #. The qualifier must begin with an alphabetic character.

Programmer response: Enter a valid qualifier and resubmit the job.

Module: BNDC002

BND601I BEGIN PROGRAM AT *day mon dd*
hh:mm:ss yyyy

Explanation: This is the date when the program began. The message shows the day of the week, the month, the date, the time, and the year that the program began.

Programmer response: None.

Module: BNDA020

BND602I END PROGRAM AT *day mon dd*
hh:mm:ss yyyy

Explanation: This is the date when the program ended. The message shows the day of the week, the month, the date, the time, and the year that the program ended.

Programmer response: None.

Module: BNDA020

BND603I nnnn SYSIBM.SYSPACKAGES
ENTRIES

Explanation: The number of entries that were found in SYSIBM.SYSPACKAGES. This number includes packages that will be automatically excluded, that is, packages that have names that begin with the letters DSN.

Programmer response: None.

Module: BNDA020

BND604I nnnn WORKING PACKAGE DATA
ENTRIES

Explanation: The number of packages that were selected from SYSIBM.SYSPACKAGES into the program's table for examination against the load libraries. The difference between the number in this message and the number in message BND603I represents packages that were excluded from examination.

Programmer response: None.

Module: BNDA020

BND605I CONNECT TO SSID *ssid* **REQUESTED**

Explanation: A DSN(*ssid*) control statement was encountered. The subsystem for which a connect was requested is identified by *ssid*.

Programmer response: None.

Module: BNDA020

BND606E SSID *ssid* **WAS NOT FOUND**

Explanation: The SSID that was coded on the DSN(*ssid*) control statement is identified by *ssid*. The program was unable to connect to the specified subsystem. The program ends with a condition code 4.

Programmer response: Correct the SSID in the control card and resubmit the job.

Module: BNDA020

BND608I REQUEST *cccc* **AS THE SET CURRENT**
SQLID

Explanation: The value in the OWNER(*cccc*) control statement is identified by *cccc*. This control statement is set as the current SQLID. The program will prefix the tables that it attempts to use with this owner ID.

Programmer response: None.

Module: BNDA020

BND611I NUMBER MUST BE FOUR BYTES.
DEFAULT VALUE OF 100 IS USED

Explanation: The number that is specified for the commit parameter must be exactly 4 digits. For example, to set the COMMIT value to 2, issue the command as COMMIT(0002); to set the COMMIT value to 9,999, issue the command as COMMIT(9999). Invalid entries are rejected and the default value of 100 is used.

Programmer response: Correct the COMMIT(*nnnn*) entry and resubmit the job. Because this information is presented at the end of the run, only make the change for runs that follow the one with the error. It is not necessary to resubmit the job that issues the message.

Module: BNDA020

BND612I REQUEST *nnnn* **AS COMMIT NUMBER**

Explanation: The number entered in the COMMIT(*nnnn*) command. That number must be in the range of 0002 and 9999 and all four digits must be coded. If they are not, the default value of 100 is used.

Programmer response: None. This is an informational message only.

Module: BNDA020

BND620E HLSFCMD DD CARD NOT FOUND

Explanation: An HLSFCMD DD statement was not present. The program is unable to write the FREE commands and will terminate.

Programmer response: Supply a valid HLSFCMD DD statement and resubmit the job.

Module: BNDA020

BND622I SYSIN DD CARD NOT FOUND

Explanation: A SYSIN DD statement was not present. The program will use the default SSID from SDSNEXIT and prefix the work table with the submitter's user ID.

Programmer response: None.

Module: BNDA020

BND623I BNDDIN FOUND

Explanation: A BNDIN DD statement was found and control parameters will be read from it.

Programmer response: None.

Module: BNDA020

BND630I BACKUPS OF THE DBRMS WILL BE WRITTEN

Explanation: The DBRMOUT DD statement was present and the program will attempt to write a backup of each DBRM for which a FREE command is written. Because of incomplete information in the DB2 catalog, it is likely that not all DBRMs will be backed up.

Programmer response: None.

Module: BNDA020

**BND635I UNABLE TO BACK UP DBRM
VERSION = NAME = COLLECTION =
MODE = VFLAG = RETURN CODE =**

Explanation: BNDA020 was unable to generate a DBRM from the catalog. This error typically occurs when the information in the catalog is incomplete, which usually occurs when there is a failure during the bind process. This error can also occur when the catalog contains residual data from earlier DB2 versions. The various data fields in the message identify the target catalog entries that could not be retrieved. Refer to the following table for an explanation of the possible return codes.

Table 21. BND635E return codes

Code	Reason
4	DBRM already existed but was replaced
8	DBRM or package not found
12	Version not found
20	Unable to allocate DBRMOUT
24	Unable to open DBRMOUT
28	No version provided (version flag = U)
32	DBRMOUT has invalid RECFM
128	GETMAIN failed for 24-bit work area (below the storage line)
136	Unable to open DSNTIAR

Programmer response: Refer to the following table for the appropriate action to take for each return code:

Table 22. BND635E return code actions

Return code	Action
4	None.
8	Specify the name of an existing DBRM package.
12	Specify a valid existing version for the package.
20	Specify a valid DBRMOUT DD statement.
24	Ensure that the DBRMOUT DD statement references a valid partitioned data set and that the user who submitted the job has update authority.
32	Specify a DBRMOUT DD statement that references a valid partitioned data set with attributes RECFM=FB and LRECL=80.
36	Specify a DBRMOUT DD statement that references a valid partitioned data set with attributes RECFM=FB and LRECL=80.
128	Resubmit the job with a larger below the line region size.
136	Verify that DSNTIAR is accessible, either through STEPLIB or the system link list, and that it is the correct version for the DB2 subsystem that is being accessed. Specify the correct subsystem name and ensure that the subsystem is up and available.
other	Any other return code indicates an unrecoverable program logic error. Save all output and contact IBM Software Support.

Module: BNDA020

BND636I BNDA021 RETURN = return

Explanation: BNDA021 searches load modules to find timestamps (consistency tokens) to determine if a package can be safely freed. A non-zero return code indicates that a problem occurred which prevented a valid search of one or more load modules. Refer to the following table for an explanation of the possible return codes.

Table 23. BND636I return codes

Code	Reason
0	Program run was successful
8	OPEN failed for diagnostic log file (//H21LOG)
12	Unable to acquire DCB storage

Table 23. BND636I return codes (continued)

Code	Reason
16	Unable to acquire buffer storage
20	Too many SYSnnn DD statements (max = 100)
24	Duplicate SYSnnn DD statement(s) encountered
28	SQL error - DECLARE CURSOR
32	SQL error - PREPARE
36	SQL error - OPEN (cursor)
40	SQL error - FETCH
44	No timestamps passed by caller (empty table)
48	Unexpected EOD reading table rows
52	SQL error - SELECT COUNT
56	Timestamp table overflow
60	Row count error
64	GETBUF failure (Binder API)
68	GETBUF failure (Binder API)
72	GETBUF failure (Binder API)
76	STARTD failure (Binder API)
80	CREATEW failure (Binder API)
84	DESERV API failure
88	RESETW failure (Binder API)
92	BLDL failure
96	INCLUDE failure (Binder API)
100	GETD error (Binder API)
104	Unable to acquire work area storage
108	SQL error - UPDATE
112	OPEN failed for LOADLIB (SYSnnn)
116	SQL error - PREPARE
120	Unable to determine DB2 version

Programmer response: A return code of 0 indicates that the program ran successfully and no action is required. A non-zero return code indicates that an internal processing error occurred that might require customer support. Save a copy of the job output, including all SYSOUT, and contact IBM Software Support.

Module: BNDA020

BND637I THIS IS A V7 SYSTEM, OR A V8 SYSTEM IN COMPATIBILITY MODETHIS IS A V8 SYSTEM

Explanation: One of the two messages is displayed depending on the subsystem that is being processed. This informational message identifies the release of DB2 for the subsystem against which BNDA020 is running

(the subsystem that is specified on the DSN statement).

Programmer response: None.

Module: BNDA020

BND690I THE FOLLOWING ARE THE SQL PA FIELDS

Explanation: An SQL error occurred.

Programmer response: Forward the complete SYSOUT to IBM Software Support.

Module: BNDA020

BND651S * ERROR *** GETBUF FAILED, CODE 1, REASON=reason code**

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND652S * ERROR *** GETBUF FAILED, CODE 2, REASON=reason code**

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND653S * ERROR *** GETBUF FAILED, CODE 3, REASON=reason code**

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND654S * ERROR *** STARTD FAILED, CODE 4, SYMPTOMS=symptoms**

Explanation: An error has occurred in the Binder API. Internal error and diagnostic information are represented by *symptoms*.

Programmer response: Ensure that each SYSnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND655S * ERROR *** CREATEW FAILED,
CODE 5, SYMPTOMS=*symptoms***

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYS*nnn* DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND656I * RUN HALTED DUE TO ERROR(S)

Explanation: The run has been halted because successful processing is impossible. This message is normally preceded by one or more error messages that describe the specific failure.

Programmer response: Correct the error and resubmit the job.

Module: BNDA006

BND657E * DBRMLIB DD STATEMENT
MISSING*****

Explanation: There is no DBRMLIB DD statement.

Programmer response: Correct the JCL and resubmit the job.

Module: BNDA006

BND658E * No load library(s) specified *****

Explanation: No LOAD*nn* DD statements were present in the execution JCL. Each load library to be scanned must be identified by a LOAD*nn* DD statement, where *nn* is a number from 00 thru 99.

Programmer response: Correct the JCL and resubmit the job.

Module: BNDA006

BND659S * UNABLE TO OPEN SYSIN *****

Explanation: SYSIN is missing, in use by another job, has incorrect attributes, or is otherwise unavailable.

Programmer response: Correct the condition and resubmit the job.

Module: BNDA006

BND660S * UNABLE TO OPEN DBRMLIB *****

Explanation: DBRMLIB is missing, in use by another job, has incorrect attributes, or is otherwise unavailable.

Programmer response: Correct the condition and resubmit the job.

Module: BNDA006

BND661E * INVALID STATEMENT *****

Explanation: The statement on the preceding line contains invalid syntax.

Programmer response: Correct the input and resubmit the job.

Module: BNDA006

BND662E * ONLY ONE SSID MAY BE
SPECIFIED *****

Explanation: Multiple SSID statements were encountered; only one is permitted.

Programmer response: Correct the input and resubmit the job.

Module: BNDA006

BND663E * OPERAND MISSING OR INVALID

Explanation: The operand on the preceding line is omitted, has invalid syntax, or is an unacceptable value.

Programmer response: Correct the input and resubmit the job.

Module: BNDA006

BND664E * INVALID NAME OR MASK
SPECIFICATION *****

Explanation: The operand on the preceding line is omitted, has invalid syntax, or is an unacceptable value.

Programmer response: Correct the input and resubmit the job.

Module: BNDA006

BND665E * SSID CONTAINS TOO MANY
CHARACTERS *****

Explanation: The subsystem ID can be a maximum of four characters in length; more than four were specified.

Programmer response: Correct the input and resubmit the job.

Module: BNDA006

BND666S * ERROR *** GETMAIN FAILED,
CODE 16, R15=*nnnn***

Explanation: Insufficient storage is available for successful processing.

Programmer response: Resubmit the job with a larger region size.

Module: BNDA006

BND667S * ERROR *** DESERV FAILED, CODE 17, SYMPTOMS= *symptoms***

Explanation: An API failure has occurred.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND668S * ERROR *** DESERV FAILED, CODE 18, SYMPTOMS= *symptoms***

Explanation: An API failure has occurred.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND669S * ERROR *** BINDER RESETW FAILED, CODE 19, SYMPTOMS= *symptoms***

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND670S * ERROR *** BLDL FAILED, CODE 20, SYMPTOMS= *symptoms***

Explanation: An API failure has occurred.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND671W * WARNING *** BINDER INCLUDE FAILED, CODE 21, SYMPTOMS= *symptoms***

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND672S * ERROR *** BINDER GETD FAILED, CODE 22, SYMPTOMS= *symptoms***

Explanation: The Binder API returned a non-zero condition code for a GETD request and that a load module is damaged or corrupted. The affected load module is ignored and processing continues with the next load module in the sequence.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND673S * ERROR *** STORAGE REQ FAILED, CODE 23, SYMPTOMS= *symptoms***

Explanation: An API failure has occurred.

Programmer response: Resubmit the job with a larger region size.

Module: BNDA006

BND674S * ERROR *** IEWBIND GETD FAILED, CODE 24, SYMPTOMS= *symptoms***

Explanation: An error has occurred in the Binder API.

Programmer response: Ensure that each SYSnnnn DD statement references a valid and accessible load library, and resubmit the job. If the problem persists, contact IBM Software Support.

Module: BNDA006

BND675S * ERROR *** LOAD FAILED FOR MODULE BNDA009**

Explanation: DBRM Checker is unable to access one of its component programs.

Programmer response: Probable installation error. Make sure that program BNDA009 is available to the job, either through a STEPLIB or through the system link list.

Module: BNDA006

BND676S * ERROR *** UNABLE TO CONNECT TO SUBSYSTEM *subsystem***

Explanation: DBRM Checker is unable to establish a connection to DB2 subsystem *subsystem*.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA006

BND677S * ERROR *** UNABLE TO ACCESS
PLAN BNDA009**

Explanation: Program BNDA009 has not been successfully bound in the DB2 subsystem you are accessing.

Programmer response: Probable installation error. Make sure that a successful Bind is run for program BNDA009 and resubmit the job.

Module: BNDA006

BND678S * ERROR *** DB2 INTERFACE
ERROR (NONZERO RETURN CODE
FROM BNDA009)**

Explanation: An error has occurred in the DB2 interface. This normally indicates an environmental error of some kind.

Programmer response: Probable installation or configuration error. Verify that DB2 Bind Manager has been properly installed.

Module: BNDA006

BND679W * WARNING *** DB2 CATALOG
WILL NOT BE VERIFIED**

Explanation: The DB2 catalog is not available. This message is normally preceded by another error message that describes the reason why the catalog cannot be accessed.

Programmer response: Correct the error condition and resubmit the job.

Module: BNDA006

BND680W * WARNING *** NO SSID
STATEMENT ENCOUNTERED**

Explanation: The SSID of the DB2 subsystem to verify DBRMs against was not specified.

Programmer response: Specify a subsystem name and resubmit the job.

Module: BNDA006

BND681I * WARNING *** LOAD MODULE
module IS BYPASSED; PROCESSING
CONTINUES**

Explanation: The load module contains invalid data, that is, a format error. This problem can indicate that the load module or the library (or both) are corrupted. The load module is ignored (skipped) and processing continues.

Programmer response: None.

Module: BNDA006

BND686W * ERROR *** Duplicate LOAD DD
statement(s)**

Explanation: One or more duplicate LOADnn DD statements were present in the execution JCL. Each load library to be scanned must be identified by a LOADnn DD statement, where nn is a number from 00 thru 99. Duplicates are not permitted; each LOADnn name must be unique within the job step.

Programmer response: Correct the JCL so that there are no duplicate DD names and resubmit the job.

Module: BNDA006

| **BND689I The DD name, *nnnnnnnn* is invalid - the
| statement is ignored**

| **Explanation:** *nnnnnnnn* is not a valid DD name for
| BNDA006. The DD statement is ignored and processing
| continues.

| **Programmer response:** None.

| **Module:** BNDA006

**BND800S UNABLE TO DEALLOCATE WORK
FILE, DIAGNOSTIC CODE=*code***

Explanation: Dynamic allocation failed to free an allocated work file. This condition should never occur in normal operation.

Programmer response: Save all output, including the ISPF log and screen shots, if available, and contact IBM Software Support.

Module: BND003

**BND801S UNABLE TO ALLOCATE WORK FILE,
DIAGNOSTIC CODE=*code***

Explanation: A temporary work file could not be allocated.

Programmer response: DB2 Bind Manager uses up to three work files of approximately five 3390 cylinders each. Ensure that sufficient work space is available to satisfy these allocations and resubmit the job. If the problem persists and sufficient work space is available, save all output, including the ISPF log and screen shots if available, and contact IBM Software Support.

Module: BND003

**BND802S UNABLE TO DEALLOCATE WORK
FILE, DIAGNOSTIC CODE=*code***

Explanation: Dynamic allocation failed to free an allocated work file. This condition should never occur in normal operation.

Programmer response: Save all output, including the ISPF log and screen shots, if available, and contact IBM Software Support.

Module: BND003

BND803E HIGH LEVEL QUALIFIER FOR WORK FILES HAS NOT BEEN SPECIFIED

Explanation: DB2 Bind Manager was unable to allocate a work file. This condition occurs when a high-level qualifier for work file data set names was not specified.

Programmer response: Select option 0 from the main menu. If the default high-level qualifier is valid, press PF3 to save it. If it is not, enter a valid qualifier and press PF3. Resubmit the job.

Module: BND003

BND804S WORK FILE ALLOCATION FAILED, S99FDBK=S99FDBK code, S99ERROR=S99ERROR code

Explanation: DB2 Bind Manager was unable to allocate a temporary work file. Refer to *MVS Programming: Authorized Assembler Services Guide* for an explanation of the codes that are returned by Dynamic Allocation (SVC 99).

Programmer response: Correct the condition that is preventing successful allocation and retry.

Module: BND003

BND805S WORK FILE ALLOCATION FAILED, S99FDBK=S99FDBK, S99ERROR=S99ERROR

Explanation: DB2 Bind Manager was unable to allocate a temporary work file. Refer to *MVS Programming: Authorized Assembler Services Guide* for an explanation of the codes that are returned by Dynamic Allocation (SVC 99).

Programmer response: Correct the condition that is preventing successful allocation and retry.

Module: BND005

BND806S WORK FILE ALLOCATION FAILED, S99FDBK=S99FDBK, S99ERROR=S99ERROR

Explanation: DB2 Bind Manager was unable to allocate a temporary work file. Refer to *MVS Programming: Authorized Assembler Services Guide* for an explanation of the codes that are returned by Dynamic Allocation (SVC 99).

Programmer response: Correct the condition that is preventing successful allocation and retry.

Module: BND006

BND2101S UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, PLAN BNDA024 RC=return code, REASON=reason code

Explanation: An unexpected DB2 error occurred. The program was trying to connect to the *subsystem* subsystem.

Programmer response: Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA024

BND2102S UNABLE TO OPEN *ddname*, REASON=reason code

Explanation: The program was unable to open *ddname ddname* because of an environmental error.

Programmer response: Verify the DD statement was specified correctly. If so, contact IBM Software Support.

Module: BNDA024

BND2104I *text*

Explanation: An input command (or comment) read from SYSIN. It is displayed for informational purposes.

Programmer response: None.

Module: BNDA024

BND2105E INVALID STATEMENT

Explanation: The input statement that precedes this message is invalid.

Programmer response: Correct or remove the invalid statement and resubmit the job.

Module: BNDA024

BND2106E MISSING OR INVALID SUBSYSTEM ID (SSID)

Explanation: The SSID statement does not specify a valid subsystem name.

Programmer response: Correct the invalid statement and resubmit the job.

Module: BNDA024

BND2107I CONNECTED TO SUBSYSTEM *subsystem*

Explanation: A connection was established to subsystem *subsystem*.

Programmer response: None. This is an informational message.

Module: BNDA024

BND2108E SSID MAY ONLY BE SPECIFIED ONCE

Explanation: More than one SSID command was specified. You can specify only one SSID command per run.

Programmer response: Correct the invalid input and resubmit the job.

Module: BNDA024

BND2110E NO SSID WAS SPECIFIED

Explanation: A valid SSID statement was not specified. The program is unable to connect to DB2.

Programmer response: Correct the invalid input and resubmit the job.

Module: BNDA024

**BND2116W ERROR IN PACKAGE STATEMENT:
MISSING OPERAND(S)**

Explanation: The PACKAGE was coded with no operands.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2121S MODULE DSNTIAR COULD NOT BE
LOADED, REASON=*reason code***

Explanation: DSNTIAR could not be accessed.

Programmer response: Verify that DSNTIAR is accessible, either through STEPLIB or the system link list.

Module: BNDA024

BND2122S DSNTIAR FAILURE, RC=*return code*

Explanation: DSNTIAR issued return code *return code*.

Programmer response: Verify that DSNTIAR is accessible either through STEPLIB or the system link list, and that it is the correct version for the DB2 subsystem that is being accessed. Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA024

**BND2125S UNABLE TO DISCONNECT FROM
SUBSYSTEM *subsystem* RC=*return code*,
REASON=*reason code***

Explanation: An unexpected DB2 error occurred. *subsystem* is the subsystem that the program was trying to disconnect from.

Programmer response: Specify the correct subsystem

name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA024

**BND2126I DISCONNECTED FROM SUBSYSTEM
*subsystem***

Explanation: This is an informational message. The connection to subsystem *subsystem* was broken.

Programmer response: None.

Module: BNDA024

**BND2129I MORE THAN ONE VERSION OF
PACKAGE *package* EXISTS; SELECT A
SPECIFIC VERSION TO PROCESS**

Explanation: More than one version is present in the subsystem, and the PACKAGE command did not specify a version.

Programmer response: Specify a valid version of the PACKAGE.

Module: BNDA024

**BND2130I PACKAGE *package* NOT FOUND IN
SUBSYSTEM *subsystem***

Explanation: The named package does not exist in the specified subsystem.

Programmer response: Specify an existing package.

Module: BNDA024

**BND2132W PACKAGE NOT FOUND IN
SUBSYSTEM *subsystem***

Explanation: The specified instance (the specific collection/package/version combination) does not exist in the specified DB2 subsystem.

Programmer response: Specify valid names for package, collection, and version.

Module: BNDA024

**BND2133I *dbrm* EXISTS IN MORE THAN ONE
PLAN; SELECT A SPECIFIC PLAN TO
PROCESS**

Explanation: More than one instance of the specified DBRM exists.

Programmer response: Identify a specific instance to process (specify a plan name).

Module: BNDA024

BND2134I DBRM *dbrm* NOT FOUND IN
SUBSYSTEM *subsystem*

Explanation: The named DBRM does not exist.

Programmer response: Specify an existing DBRM.

Module: BNDA024

BND2135E DBRM NOT FOUND IN SUBSYSTEM
subsystem

Explanation: The specified instance (the specific plan/DBRM combination) does not exist in the specified DB2 subsystem.

Programmer response: Specify valid names for plan and DBRM.

Module: BNDA024

BND2136E DBRM GENERATION FAILED,
REASON=*reason code*

Explanation: An internal processing error has occurred.

Programmer response: DB2 Bind Manager is attempting to extract DBRM information from the DB2 catalog. Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA024

BND2137I DBRM *dbrm* CREATED

Explanation: The named DBRM was generated and stored in DBRMLIB.

Programmer response: DB2 Bind Manager is attempting to extract DBRM information from the DB2 catalog. Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA024

BND2138E PACKAGE GENERATION FAILED,
REASON=*reason code*

Explanation: An internal processing error has occurred.

Programmer response: Contact IBM Software Support.

Module: BNDA024

BND2139I PACKAGE *package* CREATED

Explanation: The named package has been re-created (as a DBRM) and stored in DBRMLIB.

Programmer response: None.

Module: BNDA024

BND2142E *ddname* DD STATEMENT MISSING

Explanation: The DD statement *ddname* is not present.

Programmer response: Supply the missing DD statement and resubmit the job.

Module: BNDA024

BND2143E UNABLE TO OPEN DBRMLIB,
REASON=*reason code*

Explanation: DBRMLIB could not be opened.

Programmer response: Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes, for example, RECFM=FB and LRECL=80.

Module: BNDA024

BND2144I DBRM *dbrm* REPLACED

Explanation: The named DBRM was generated and stored in DBRMLIB.

Programmer response: None.

Module: BNDA024

BND2145I PACKAGE *package* REPLACED

Explanation: The named package was re-created (as a DBRM) and stored in DBRMLIB.

Programmer response: None.

Module: BNDA024

BND2146S BPAM API FAILURE,
FUNCTION=WRITE,
SYMPTOM=*symptom*

Explanation: An internal processing error has occurred.

Programmer response: Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes, for example, RECFM=FB and LRECL=80.

Module: BNDA024

BND2147S BPAM API FAILURE,
FUNCTION=WRITE,
SYMPTOM=*symptom*

Explanation: An internal processing error has occurred.

Programmer response: Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes, for example, RECFM=FB and LRECL=80.

Module: BNDA024

BND2149E *keyword* IS NOT A VALID KEYWORD FOR THE LIST COMMAND

Explanation: A syntax error exists in the LIST command.

Programmer response: Correct the command and resubmit the job.

Module: BNDA024

BND2150E DEFAULTS MAY ONLY BE SPECIFIED ONCE

Explanation: Multiple DEFAULTS commands were encountered.

Programmer response: Remove all but one of the DEFAULTS commands.

Module: BNDA024

BND2151W NO OPERANDS SPECIFIED ON DEFAULTS STATEMENT

Explanation: No operands were specified on the DEFAULTS statements.

Programmer response: Specify valid operands or remove the command.

Module: BNDA024

BND2152W DEFAULTS STATEMENT HAS INVALID SYNTAX

Explanation: The syntax in the DEFAULTS statement is invalid.

Programmer response: Correct the command or remove it.

Module: BNDA024

BND2153W DEFAULTS STATEMENT HAS INVALID SYNTAX

Explanation: The syntax in the DEFAULTS statement is invalid.

Programmer response: Correct the command or remove it.

Module: BNDA024

BND2156W *operand* IS NOT A VALID OPERAND FOR THIS STATEMENT. THE STATEMENT IS IGNORED.

Explanation: The operand *operand* is not valid for the command.

Programmer response: Correct the invalid operand or remove the command.

Module: BNDA024

BND2157W DEFAULTS STATEMENT DOES NOT SPECIFY A VALID LANGUAGE. THE STATEMENT IS IGNORED.

Explanation: The language that was specified in the DEFAULTS statement is invalid.

Programmer response: Specify a valid language or remove the command.

Module: BNDA024

BND2158W DEFAULTS STATEMENT DOES NOT SPECIFY A VALID DB2 VERSION. THE STATEMENT IS IGNORED.

Explanation: The DB2 version that was specified in the DEFAULTS statement is invalid.

Programmer response: Specify a valid version or remove the command.

Module: BNDA024

BND2159W *operand* IS NOT A VALID OPERAND FOR THIS STATEMENT. THE STATEMENT IS IGNORED.

Explanation: The operand *operand* is not valid for the command.

Programmer response: Correct the invalid operand or remove the command.

Module: BNDA024

BND2160W *operand* IS NOT A VALID OPERAND FOR THIS STATEMENT. THE STATEMENT IS IGNORED.

Explanation: The operand *operand* is not valid for the command.

Programmer response: Correct the invalid operand or remove the command.

Module: BNDA024

BND2161W WARNING - ONE OR MORE GENERATED DBRMS ARE MISSING DB2 VERSION AND/OR HOST LANGUAGE VALUES, AND NO DEFAULT VALUES WERE SUPPLIED. THE GENERATED DBRM MAY NOT BE USABLE.

Explanation: DBRMS were generated but might not be valid. A DEFAULTS command was not present to supply the missing values.

Programmer response: Add a DEFAULTS command and resubmit the job.

Module: BNDA024

BND2162E NO DB2 VERSION WAS SPECIFIED.

Explanation: A value was not specified for the DB2= keyword.

Programmer response: Supply a value and resubmit the job.

Module: BNDA024

BND2163E NO LANGUAGE WAS SPECIFIED

Explanation: A value was not specified for the LANGUAGE= keyword.

Programmer response: Supply a value and resubmit the job.

Module: BNDA024

**BND2164E ERROR IN DBRM STATEMENT:
MISSING OPERAND(S)**

Explanation: A DBRM command was specified with no operands.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2165E ERROR IN DBRM STATEMENT:
MISSING DBRM NAME**

Explanation: A DBRM command was specified with a plan name but no DBRM name.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2166E ERROR IN PACKAGE STATEMENT:
MISSING PACKAGE NAME**

Explanation: A PACKAGE command was specified with a collection name but no package name.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2167E ERROR IN PACKAGE STATEMENT:
MISSING VERSION**

Explanation: A PACKAGE command was specified with a collection name and package but no version ID.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2168E ERROR IN PACKAGE STATEMENT:
SYNTAX -- VERSION**

Explanation: A PACKAGE command was specified with an invalid version ID.

Programmer response: Correct the statement and resubmit the job.

Module: BNDA024

**BND2171S INTERNAL PROCESSING ERROR --
CALL TECHNICAL SUPPORT**

Explanation: A program logic error was detected. This condition should not occur in normal processing.

Programmer response: Save all output from the entire job (not only the BNDA024 job step) and contact IBM Software Support.

Module: BNDA024

**BND2172S UNEXPECTED END-OF-DATA
ENCOUNTERED READING DBRM
TEXT**

Explanation: End of data occurred prematurely while DBRM data was being read from the DB2 catalog.

Programmer response: Contact IBM Software Support.

Module: BNDA024

**BND2173W WARNING: UNABLE TO REFORMAT
HOST VARIABLE PLACEHOLDER IN
STATEMENT *statement*, SECTION
section, AT COLUMN *column***

Explanation: BNDA024 was unable to insert the optimal number of blanks into the generated SQL text. This is normally not an error.

Programmer response: None.

Module: BNDA024

**BND2177I DB2 RELEASES EARLIER THAN 2.3
ARE NOT SUPPORTED; *package* WAS
NOT GENERATED**

Explanation: BNDA024 does not support DBRM generation for packages or plans that were created for DB2 version 2.2 or earlier. The specified package or DBRM is not produced and processing continues with the next request, if any.

Programmer response: None.

Module: BNDA024

BND2199S ERROR IN BNDA024 - SYMPTOM=*symptom*

Explanation: A program logic error was detected.

Programmer response: BNDA024 is attempting to display a nonexistent message. This error is probably a problem logic error. Save all output and contact IBM Software Support.

Module: BNDA024

BND2201S UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, PLAN BNDA030RC=*return code*, REASON=*reason code*

Explanation: An unexpected DB2 error occurred. The program was trying to connect to the *subsystem* subsystem.

Programmer response: Specify the correct subsystem name on the SSID statement, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA030

BND2202S UNABLE TO OPEN *ddname*, REASON=*reason code*

Explanation: The program was unable to open ddname *ddname* because of an environmental error.

Programmer response: Verify the DD statement was specified correctly. If so, contact IBM Software Support.

Module: BNDA030

BND2203E ERROR IN STATEMENT - NO DBRM NAME SPECIFIED

Explanation: The DBRM command was issued without an operand.

Programmer response: Correct the invalid statement and resubmit the job.

Module: BNDA030

BND2204I *text*

Explanation: An input command (or comment) read from SYSIN. It is displayed for informational purposes.

Programmer response: None.

Module: BNDA030

BND2205E INVALID STATEMENT

Explanation: The input statement that precedes this message is invalid.

Programmer response: Correct or remove the invalid statement and resubmit the job.

Module: BNDA030

BND2206E MISSING OR INVALID SUBSYSTEM ID (SSID)

Explanation: The SSID statement does not specify a valid subsystem name.

Programmer response: Correct the invalid statement and resubmit the job.

Module: BNDA030

BND2207I CONNECTED TO SUBSYSTEM *subsystem*

Explanation: This is an informational message. A connection was established to subsystem *subsystem*.

Programmer response: None.

Module: BNDA030

BND2208E SSID MAY ONLY BE SPECIFIED ONCE

Explanation: More than one SSID command was specified. You can specify only one SSID command per run.

Programmer response: Correct the invalid input and resubmit the job.

Module: BNDA030

BND2209E ERROR IN STATEMENT - INVALID DBRM NAME

Explanation: The DBRM name or pattern is longer than eight characters.

Programmer response: Correct the invalid statement and resubmit the job.

Module: BNDA030

BND2210E NO SSID WAS SPECIFIED

Explanation: A valid SSID statement was not specified. The program is unable to connect to DB2.

Programmer response: Correct the invalid input and resubmit the job.

Module: BNDA030

BND2211E SYNTAX ERROR - DBRM NAME

Explanation: The specified DBRM name or pattern is not valid.

Programmer response: Correct the invalid input and resubmit the job.

Module: BNDA030

**BND2212S UNABLE TO OPEN DBRMLIB,
REASON=*reason code***

Explanation: The program was unable to open DBRMLIB. A reason code, *reason code*, provides diagnostic information for technical support.

Programmer response: Specify a valid DBRMLIB DD statement, ensure that the DD statement refers to a valid existing library, and resubmit the job.

Module: BNDA030

BND2213E API ERROR, SYMPTOM=*symptom*

Explanation: An error occurred reading from DBRMLIB.

Programmer response: Specify a valid DBRMLIB DD statement, ensure that the DD statement refers to a valid existing library, and resubmit the job.

Module: BNDA030

BND2214E API ERROR, SYMPTOM=*symptom*

Explanation: An error occurred reading from DBRMLIB.

Programmer response: Specify a valid DBRMLIB DD statement, ensure that the DD statement refers to a valid existing library, and resubmit the job.

Module: BNDA030

BND2215E API ERROR, SYMPTOM=*symptom*

Explanation: An error occurred reading from DBRMLIB.

Programmer response: Specify a valid DBRMLIB DD statement, ensure that the DD statement refers to a valid existing library, and resubmit the job.

Module: BNDA030

BND2216E API ERROR, SYMPTOM=*symptom*

Explanation: An error occurred reading from DBRMLIB.

Programmer response: Specify a valid DBRMLIB DD statement, ensure that the DD statement refers to a valid existing library, and resubmit the job.

Module: BNDA030

**BND2221S MODULE DSNTIAR COULD NOT BE
LOADED, REASON=*reason code***

Explanation: DSNTIAR could not be accessed. Internal error and diagnostic information are represented by *reason code*.

Programmer response: Make certain that DSNTIAR is

accessible, either through STEPLIB or the system link list.

Module: BNDA030

BND2222S DSNTIAR FAILURE, RC=*return code*

Explanation: DSNTIAR issued return code *return code*.

Programmer response: Verify that DSNTIAR is accessible, either through STEPLIB or the system link list, and that it is the correct version for the DB2 subsystem that is being accessed. Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA030

**BND2225S UNABLE TO DISCONNECT FROM
SUBSYSTEM *subsystem* RC=*return code*,
REASON=*reason code***

Explanation: An unexpected DB2 error occurred. The program was trying to disconnect from the *subsystem* subsystem.

Programmer response: Specify the correct subsystem name on the SSID statement, ensure that the subsystem is up and available, and resubmit the job.

Module: BNDA030

**BND2226I DISCONNECTED FROM SUBSYSTEM
*subsystem***

Explanation: This is an informational message. The connection to subsystem *subsystem* was terminated.

Programmer response: None.

Module: BNDA030

BND2242E *ddname* DD STATEMENT MISSING

Explanation: The DD statement named *ddname* is not present.

Programmer response: Supply the missing DD statement and resubmit the job.

Module: BNDA030

**BND2243I *dbrm* IS AN EMPTY DBRM AND WAS
BYPASSED**

Explanation: The specified DBRM is empty (a directory entry exists but no data is present).

Programmer response: None.

Module: BNDA030

BND2244E TOO MANY DBRMS SELECTED

Explanation: The DBRM command caused more than 1000 DBRMs to be selected for processing. A single command can process a maximum of 1000 DBRMs.

Programmer response: Refine your search criteria either by using fewer wild card characters or a different name, and resubmit the job.

Module: BNDA030

BND2299S ERROR IN BNDA030 - SYMPTOM=*symptom*

Explanation: A program logic error was detected.

Programmer response: BNDA030 is attempting to display a nonexistent message. This error is probably a problem logic error. Save all output and contact IBM Software Support.

Module: BNDA030

BND2301I * ERROR *** BINDER INCLUDE FAILED, CODE 22, SYMPTOMS=*symptoms***

Explanation: The Binder API returned a non-zero condition code for an INCLUDE request and that a load module is damaged or corrupted. This message is normally accompanied by message BND2306W.

Programmer response: None.

Module: BNDA007

BND2302I * ERROR *** BINDER GETN FAILED, CODE 22, SYMPTOMS=*symptoms***

Explanation: The Binder API returned a non-zero condition code for a GETN request and that a load module is damaged or corrupted. This message is normally accompanied by message BND2306W.

Programmer response: None.

Module: BNDA007

BND2303I * ERROR *** BINDER GETN FAILED, CODE 22, SYMPTOMS=*symptom***

Explanation: The Binder API returned a non-zero condition code for a GETN request and that a load module is damaged or corrupted. This message is normally accompanied by message BND2306W.

Programmer response: None.

Module: BNDA007

BND2304I * ERROR *** BINDER GETE FAILED, CODE 22, SYMPTOMS=*symptoms***

Explanation: The Binder API returned a non-zero condition code for a GETE request and that a load module is damaged or corrupted. This message is normally accompanied by message BND2306W.

Programmer response: None.

Module: BNDA007

BND2305I * ERROR *** BINDER GETE FAILED, CODE 22, SYMPTOMS=*symptoms***

Explanation: The Binder API returned a non-zero condition code for a GETE request and that a load module is damaged or corrupted. This message is normally accompanied by message BND2306W.

Programmer response: None.

Module: BNDA007

BND2306W * ERROR *** *** WARNING *** LOAD MODULE *module* BYPASSED, PROCESSING CONTINUES**

Explanation: Load module *module* is damaged or corrupted. It is preceded by message BND2301I, BND2302I, BND2303I, BND2304I, or BND2305I. This load module is ignored and processing continues with the next load module in the sequence.

Programmer response: None.

Module: BNDA007

BND2401S *ddname* DD STATEMENT MISSING

Explanation: A DD statement was not supplied for *ddname ddname*.

Programmer response: Add the missing DD statement to the JCL and resubmit the job.

Module: BNDA036

BND2402S UNABLE TO OPEN *ddname*, REASON=*reason code*

Explanation: OPEN failed for the data set that was associated with *ddname ddname*. Refer to *Enterprise PL/I for z/OS Messages and Codes* for an explanation of the reason code.

Programmer response: Correct the condition that is preventing a successful open and resubmit the job.

Module: BNDA036

BND2403S MODULE *module* COULD NOT BE LOADED, REASON=*reason code*

Explanation: FETCH failed for program (load module) *module*. A common cause of this error is a missing STEPLIB in the JCL. Refer to *Enterprise PL/I for z/OS Messages and Codes* for an explanation of the reason code.

Programmer response: Correct the condition and resubmit the job.

Module: BNDA036

BND2404E NO SUBSYSTEM NAME HAS BEEN SPECIFIED

Explanation: A DB2 subsystem name was not specified. DB2 Bind Manager does not know which DB2 subsystem to access.

Programmer response: Code the SSID command with a valid subsystem name and resubmit the job.

Module: BNDA036

BND2405E INVALID COMMAND

Explanation: The command that was entered is not recognized or has invalid syntax.

Programmer response: Correct or remove the invalid command and resubmit the job.

Module: BNDA036

BND2406E SSID MAY ONLY BE SPECIFIED ONCE

Explanation: Multiple SSID commands were entered. You can issue only one SSID command per run.

Programmer response: Remove the superfluous commands and resubmit the job.

Module: BNDA036

BND2407E MISSING OR INVALID SSID

Explanation: The SSID command was issued without an operand, or the operand is invalid.

Programmer response: Correct the SSID command and resubmit the job.

Module: BNDA036

BND2408E UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, RC=*return code*, REASON=*reason code*

Explanation: DB2 Bind Manager is unable to establish a connection to DB2 subsystem *subsystem*. Refer to *DB2 Application Programming and SQL Guide*, for an explanation of the return code and reason code.

Programmer response: Verify that the correct

subsystem name was coded on the SSID statement and that the subsystem is up and available. Correct the condition that is preventing the connection and resubmit the job.

Module: BNDA036

BND2409S INVALID DATA IN SYSIBM.SYSCOLUMNS, SYMPTOM=*symptom*

Explanation: The DB2 catalog contains invalid data. The catalog might be corrupted. This condition should never occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDA036

BND2410E UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, BIND REQUIRED

Explanation: DB2 Bind Manager is unable to establish a connection to DB2 subsystem *subsystem* using plan BNDA036. The plan has not been bound in this subsystem, or the bind is out of date.

Programmer response: Verify that the correct subsystem name was coded on the SSID statement and that the bind for plan BNDA036 is current. If necessary, bind the plan. Resubmit the job.

Module: BNDA036

BND2411I CONNECTED TO SUBSYSTEM *subsystem*

Explanation: DB2 Bind Manager has successfully connected to DB2 subsystem *subsystem*.

Programmer response: None.

Module: BNDA036

BND2412I DISCONNECTED FROM *subsystem*

Explanation: DB2 Bind Manager has disconnected from DB2 subsystem *subsystem*.

Programmer response: None.

Module: BNDA036

BND2413I HOST LANGUAGE INDETERMINATE, PACKAGE SKIPPED. PACKAGE *package*, CONTOKEN *contoken*.

Explanation: Based on the information available from the catalog, Bind Manager is unable to determine the host language for the package. The package is ignored and processing continues for the remaining packages.

Programmer response: None.

Module: BNDA036

BND2414S GETMAIN FAILED IN JFCB ROUTINE

Explanation: DB2 Bind Manager was unable to acquire sufficient working storage.

Programmer response: Increase the region size and resubmit the job.

Module: BNDA036

BND2415S DSNTIAR FAILURE, RC=return code

Explanation: DSNTIAR issued a non-zero condition code. The condition code that was issued by DSNTIAR is indicated by *return code*.

Programmer response: Verify that the version of DSNTIAR that is used is the correct version for the version of DB2 that is running in the specified subsystem. DSNTIAR is accessed through STEPLIB or, if not found there, through the system link list.

Module: BNDA036

BND2416W UNABLE TO PROCESS *dsname* (INVALID RECFM). BYPASSED.

Explanation: The data set that is referenced by *dsname* is not a load library; the data set does not have RECFM=U. The data set is not processed.

Programmer response: Specify a valid load library name.

Module: BNDA036

BND2417W UNABLE TO PROCESS *dsname* (NOT PARTITIONED). BYPASSED.

Explanation: The data set that is referenced by *dsname* is not a PDS. The data set is not processed.

Programmer response: Specify a valid load library name and resubmit the job.

Module: BNDA036

BND2418W UNABLE TO OPEN *dsname*. BYPASSED.

Explanation: The data set that is referenced by *dsname* could not be opened for input. The data set is not processed.

Programmer response: Verify that the data set exists, is not empty, and is not in use by another job or user. Resubmit the job.

Module: BNDA036

BND2419E DESERV ERROR, UNABLE TO READ DIRECTORY FOR *dsname*

Explanation: The directory for data set *dsname* could not be accessed.

Programmer response: Verify that the data set exists, is not empty, and is not in use by another job or user. Resubmit the job.

Module: BNDA036

BND2420S BINDER INITIALIZATION FAILED, RC1=return code 1, RC2=return code 2, REASON=reason code, COUNTER=counter

Explanation: The Binder API, which is used to retrieve load modules for scanning, failed to initialize properly. RC2 and reason are codes from the Binder API. RC1 and counter are internal BNDA036 trace codes that are used for diagnostic purposes. This condition should never occur in normal operation.

Programmer response: Save all output, including the message text, and contact IBM Software Support.

Module: BNDA036

BND2421I BINDER LOAD FAILED, RC1= return code 1, RC2=return code 2, REASON=reason code, COUNTER=counter LOAD MODULE loadmodule BYPASSED

Explanation: The Binder was unable to retrieve a load module for scanning. The defective load module is bypassed and scanning continues with the next load module in sequence. RC2 and reason are codes from the Binder API and are described in *MVS Program Management: Advanced Facilities*. RC1 and counter are internal BNDA036 trace codes that are used for diagnostic purposes.

Programmer response: None.

Module: BNDA036

BND2422I BINDER GETN FAILED, RC1= return code 1, RC2=return code 2, REASON=reason code, COUNTER=counter LOAD MODULE loadmodule BYPASSED

Explanation: The Binder was unable to retrieve a load module for scanning. The defective load module is bypassed and scanning continues with the next load module in sequence. RC2 and reason are codes from the Binder API and are described in *MVS Program Management: Advanced Facilities*. RC1 and counter are internal BNDA036 trace codes that are used for diagnostic purposes.

Programmer response: None.

Module: BNDA036

BND2423I NOW PROCESSING *dsname*

Explanation: Identifies the load library that is currently being processed.

Programmer response: None.

Module: BNDA036

BND2424I UNABLE TO RETRIEVE OBJECT TEXT FOR MODULE *module*, CSECT *csect*, RC=*return*, REASON=*reason*, COUNTER=*counter* CSECT *csect* BYPASSED

Explanation: The Binder was unable to retrieve text from a load module. The defective load module is bypassed and scanning continues with the next load module in sequence. RC2 and reason are codes from the Binder API and are described in *MVS Program Management: Advanced Facilities*. Counter is an internal BNDA036 trace code that is used for diagnostic purposes.

Programmer response: None.

Module: BNDA036

BND2425I PACKAGE BYPASSED DUE TO LEVEL PRECOMPILER OPTIONName=*name* Version=*version*

Explanation: The BNDA036 program encountered a package that was precompiled by using the LEVEL option. BNDA036 does not support the LEVEL precompiler option. This message is issued only once per unique package name or version, regardless of how many collections that contain the package. The version is displayed when the package has a version ID.

Programmer response: None.

Module: BNDA036

BND2426I *nnnn* PACKAGES WERE BYPASSED DUE TO LEVEL OPTION

Explanation: The number (*nnnn*) of packages in the catalog that were not processed because the LEVEL precompiler option was used.

Programmer response: None. This is an informational message only.

Module: BNDA036

BND2427I PACKAGE *package* IS USED BY MODULE *module*, CSECT *csect*; CONTOKEN FOUND AT OFFSET *offset* (HEX) (VERSION=*version*)

Explanation: A consistency token in a load module

was found. The consistency token matches one from a package in the subsystem. This condition indicates that the package is in use by one or more load modules, therefore, a FREE command will not be generated for this package. The version number is displayed when the package has a version ID.

Programmer response: None.

Module: BNDA036

BND2428E NO VALUE SPECIFIED FOR COLLID

Explanation: The COLLID command was entered without an operand.

Programmer response: Add a collection name or pattern to the failing command and resubmit the job.

Module: BNDA036

BND2429E TOO MANY COLLID COMMANDS

Explanation: More than 25 COLLID commands were entered.

Programmer response: Remove the excess commands and resubmit the job.

Module: BNDA036

BND2430E UNSUPPORTED RELEASE OF DB2 – FOR VERSION 7 AND EARLIER, USE BNDA046

Explanation: BNDA036 supports DB2 Version 8 and later only. For version 7 and earlier, use the BNDA046 program.

Programmer response: Resubmit the job using BNDA046.

Module: BNDA036

BND2431E UNSUPPORTED RELEASE OF DB2 — FOR VERSION 8 AND LATER, USE BNDA036

Explanation: BNDA046 supports only DB2 Version 7 and earlier. For DB2 Version 8 and later, use BNDA046.

Programmer response: Resubmit the job using BNDA036.

Module: BNDA046

BND2434E NO VALUE SPECIFIED FOR PACKAGE.

Explanation: The PACKAGE command was specified without any package names.

Programmer response: Code the command with a package name or range and resubmit.

Module: BNDA036

BND2448I PACKAGE *NNN* IN COLLECTION *CCC*
IS MARKED "INOPERATIVE" AND
WILL BE FREED

Explanation: The named package is identified as inoperative in the DB2 catalog. This message is issued when using dynamic allocation in DEV mode. A FREE command will be generated for the package.

Programmer response: None.

Module: BNDA036

BND2449I PACKAGE *NNN* IN COLLECTION *CCC*
IS MARKED "INVALID"

Explanation: The named package is identified as invalid in the DB2 catalog. This message is issued when using dynamic allocation in PROD mode.

Programmer response: None.

Module: BNDA036

BND2450I PACKAGE *NNN* IN COLLECTION *CCC*
IS MARKED "INVALID", IS MORE
THAN 6 MONTHS OLD, AND WILL
BE FREED

Explanation: The named package is identified as invalid in the DB2 catalog. This message is issued when using dynamic allocation in DEV mode. A FREE command will be generated for the package.

Programmer response: None.

Module: BNDA036

BND2451I UNABLE TO DETERMINE AGE OF
INVALID PACKAGE *NNN* IN
COLLECTION *CCC*

Explanation: Due to invalid data in the catalog, Bind Manager is unable to calculate the age of package *nnn*. This message is accompanied by message BND2452I.

Programmer response: None.

Module: BNDA036

BND2452I PACKAGE WILL NOT BE FREED

Explanation: Informational. This message accompanies message BND2451I.

Programmer response: None.

Module: BNDA036

BND2453I MODULE *module name* could not be
loaded, reason=*reason code*

Explanation: Fetch was unable to load module *module name* into storage.

Programmer response: Although DB2 Bind Manager

reports this error, this error is detected by the operating system. When this message is issued, the job log will also contain an IEW message that describes the specific problem. See *z/OS Messages and Codes* for more information about this error. If the problem persists, contact IBM Software Support with the reason code that is indicated in the message text.

Module: BNDA036

BND2454I Package *pppppp* in collection *ccccc* is a
trigger.

Explanation: The indicated package is a trigger package. No FREE command will be generated for it, regardless of any other conditions. In the message text, *pppppp* is the package name and *ccccc* is the collection in which the package is bound.

Programmer response: None.

Module: BNDA036

BND2499S ERROR IN BNDA036 -
SYMPTOM=*symptom*

Explanation: An interface error has occurred in program BNDA036. This condition should never occur in normal operation.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDA036

BND2461E Invalid command (RESTART may not
be combined with other commands)

Explanation: When the RESTART command is used, it must be the only specified command. It may not be used in the same run with any other commands.

Programmer response: If the job is intended to be the first job of a series, remove the RESTART command, specify the command and options, and submit. If the job is not the first job in a series, remove the other commands, add the RESTART command, and submit until the job has completed processing.

Module: BNDA036

BND2465S Premature end of data in checkpoint file

Explanation: The checkpoint file has been overwritten or corrupted.

Programmer response: Rerun the job series, starting with the initial job.

Module: BNDA036

BND2466I Run complete -- commands generated

Explanation: After the final job of a Checkpoint/Restart series, this message indicates that all requested packages and load modules have been checked and FREE commands, if any, have been generated.

Programmer response: No action is required.

Module: BNDA036

BND2467I Job series is complete -- RESTART is ignored

Explanation: The final RESTART job of the series is complete, and any FREE commands have been issued.

Programmer response: No action is required.

Module: BNDA036

BND2701S Module xxxxxxxx could not be loaded.

Explanation: DB2 Bind Manager was unable to access a needed load module. In the message text, xxxxxxxx is the name of the unavailable module.

Programmer response: Make the specified load module available to the job, by using either JOBLIB/STEPLIB or the system link list.

Module: BNDA043

BND2703S Unable to connect to subsystem ssss, plan BNDA043. RC=xxxxxxx, REASON=yyyyyyyy.

Explanation: The attempt to connect to DB2 by using Call Attach failed. In the message text, ssss is the subsystem, xxxxxxxx is the Call Attach return code, and yyyyyyyy is the reason code.

Programmer response: Make sure the correct subsystem name was specified on the SSID command and that the indicated subsystem is running and available.

Module: BNDA043

BND2705S DSNTIAR failure, RC=xxxxxxx.

Explanation: Diagnostic routine DSNTIAR issued a nonzero return code (xxxxxxx in the message).

Programmer response: Make sure that DSNTIAR is accessible to the job by using either JOBLIB/STEPLIB or the system link list. Also, ensure that DSNTIAR is the correct version for the DB2 release that is being used.

Module: BNDA043

BND2706S A runtime error occurred in xxxxxxxx, symptom=yyyyyyyy.

Explanation: An internal processing error in program logic occurred. In the message text, xxxxxxxx and yyyyyyyy are diagnostic codes for use by IBM Software Support.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDA043

BND2707S The dddddddd DD statement is missing.

Explanation: A required DD statement is missing from the execution JCL. In the message text, dddddddd is the ddname of the required statement.

Programmer response: Supply the missing DD statement in the JCL and rerun the job.

Module: BNDA043

BND2708S Unable to open xxxxxxxx, reason=yyyyyyyy

Explanation: OPEN failed for the indicated data set. In the message text, xxxxxxxx is the ddname of the data set that failed to open, and yyyyyyyy is the code that was returned by OPEN to describe the type of failure. The code that is represented by yyyyyyyy is a PL/I oncode. The individual codes and their meanings are documented in *PL/I Messages and Codes*, SC27-1461.

Programmer response: Correct the condition that caused the OPEN failure and rerun the job.

Module: BNDA043

BND2710S Unable to disconnect from subsystem ssss. RC=xxxxxxx, REASON=yyyyyyyy.

Explanation: The attempt to connect to DB2 by using Call Attach was not successful. In the message text, ssss is the subsystem, xxxxxxxx is the Call Attach return code, and yyyyyyyy is the reason code.

Programmer response: This message is typically issued to describe a condition that occurs at end-of-job, after all requested DBRMs and packages have been processed. Verify that the expected output has been generated.

Module: BNDA043

BND2712W An invalid command was issued.

Explanation: A command was entered that is either unrecognizable or has incorrect syntax. The command is ignored and processing continues. This message is preceded by message BND2704I, which shows the rejected command.

| **Programmer response:** Correct the invalid command and rerun the job.

| **Module:** BNDA043

| **BND2713W The subsystem ID is missing.**

| **Explanation:** The SSID command was issued without a subsystem name. The command is ignored and processing continues. This message is preceded by message BND2704I, which shows the rejected command.

| **Programmer response:** Correct the invalid command and rerun the job.

| **Module:** BNDA043

| **BND2714W An invalid subsystem ID command (for length) was issued.**

| **Explanation:** The SSID command was issued with a subsystem name that is longer than four characters. The command is ignored and processing continues. This message is preceded by message BND2704I, which shows the rejected command.

| **Programmer response:** Correct the invalid command and rerun the job.

| **Module:** BNDA043

| **BND2716W An invalid DISPLAY command was issued.**

| **Explanation:** A DISPLAY command was issued with incorrect syntax. The command is ignored and processing continues. This message is preceded by message BND2704I, which shows the rejected command.

| **Programmer response:** Correct the invalid command and rerun the job.

| **Module:** BNDA043

| **BND2717W An invalid DISPLAY DBRM command (for length) was issued.**

| **Explanation:** A DISPLAY DBRM command was issued with a DBRM name longer than eight characters. The command is ignored and processing continues. This message is preceded by message BND2704I, which shows the rejected command.

| **Programmer response:** Correct the invalid command and rerun the job.

| **Module:** BNDA043

| **BND2718S An invalid DBRMLIB (RECFM) was referenced.**

| **Explanation:** The data set that is referenced by ddname DBRMLIB is not a valid DBRM library; it does not have the correct record format (F or FB).

| **Programmer response:** Point DBRMLIB to a valid DBRM library.

| **Module:** BNDA043

| **BND2719S An invalid DBRMLIB (LRECL) is referenced.**

| **Explanation:** The data set that is referenced by ddname DBRMLIB is not a valid DBRM library; it has a record length (LRECL) other than 80.

| **Programmer response:** Point DBRMLIB to a valid DBRM library.

| **Module:** BNDA043

| **BND2720S The DBRMLIB is invalid (not partitioned).**

| **Explanation:** The data set that is referenced by ddname DBRMLIB is not a partitioned data set (PDS).

| **Programmer response:** Point DBRMLIB to a valid DBRM library.

| **Module:** BNDA043

| **BND2721S DB2 Bind Manager was unable to open DBRMLIB.**

| **Explanation:** The data set that is referenced by ddname DBRMLIB could not be opened.

| **Programmer response:** Make sure that DBRMLIB points to a valid DBRM library. Check the job log for system messages that might indicate the reason for the OPEN failure. Correct the condition and rerun the job.

| **Module:** BNDA043

| **BND2722W Member xxxxxxxx was not found in DBRMLIB.**

| **Explanation:** The DBRM name that is specified on a DISPLAY DBRM command does not exist in the referenced DBRMLIB.

| **Programmer response:** Make sure that the correct data set is named on the DBRMLIB DD statement and that the name of an existing member is specified on the DISPLAY DBRM command.

| **Module:** BNDA043

BND2723S **BNDA043 was unable to open SYSUT1, reason=xxxxxxx.**

Explanation: BNDA043 was unable to open its work file. In the message text, xxxxxxxx is the return code from OPEN.

Programmer response: Make sure the JCL contains a valid SYSUT1 DD statement. Check the job log for system messages that might indicate the reason for the OPEN failure. Correct the condition and rerun the job.

Module: BNDA043

BND2724S **A BPAM API failure occurred, symptom= xxxxxxxx.**

Explanation: BNDA043 experienced an internal processing error. In the message text, xxxxxxxx is an internal diagnostic code for use by IBM Software Support.

Programmer response: Save all output. Contact IBM Software Support

Module: BNDA043

BND2725S **The Unicode conversion failed for dddddd, symptoms=xxxxxxxx/yyyyyyyy.**

Explanation: Unicode conversion was unsuccessful while processing a DBRM or package. In the message text, dddddd is the name of the DBRM, xxxxxxxx is the return code from Unicode System Services, and yyyyyyyy is the reason code.

Programmer response: Inspect the DBRM to determine the correct character set ID. Make sure that Unicode System Services on your system is configured to convert that character set correctly.

Module: BNDA043

BND2726S **There is no active DB2 connection and the command was ignored.**

Explanation: A DISPLAY PACKAGE command was issued but no DB2 connection is active.

Programmer response: Add an SSID command to the job to establish a connection to the DB2 subsystem that contains the package to be displayed. The SSID command must be specified before the DISPLAY PACKAGE command.

Module: BNDA043

BND2727E **An invalid command was issued, a package name is missing.**

Explanation: A DISPLAY PACKAGE command was issued without a package name.

Programmer response: Provide a package name in the command and rerun the job.

Module: BNDA043

BND2728E **An invalid command was issued, the version ID is missing.**

Explanation: A DISPLAY PACKAGE command was issued with a package and collection name but no version. The version ID is required. If the version is blank, code a pair of parentheses.

Programmer response: Provide the version ID in the command and rerun the job.

Module: BNDA043

BND2729E **A syntax error occurred, the version ID is invalid.**

Explanation: A DISPLAY PACKAGE command was issued in which the version ID is invalid. Either the leading or trailing parenthesis is missing.

Programmer response: Correct the command by inserting a parenthesis and rerun the job.

Module: BNDA043

BND2730W **Package pppppppp was not found in DB2 Subsystem ssss.**

Explanation: A DISPLAY PACKAGE command was issued for package pppppppp, but the package was not present in DB2 subsystem ssss.

Programmer response: Make sure that both the package name on the DISPLAY command and the subsystem ID on the SSID command are correct.

Module: BNDA043

BND2731S **DBRM extraction failed with condition code= xxxxxxxx.**

Explanation: The package that was specified on a DISPLAY PACKAGE statement could not be retrieved from the DB2 catalog due to an internal processing error.

Programmer response: Save all output. Contact IBM Software Support.

Module: BNDA043

BND2732S **Unable to access JFCB for DBRMLIB.**

Explanation: The Job File Control Block for the data set that is referenced by ddname DBRMLIB could not be retrieved even though the DD statement is present.

Programmer response: Save all output. Contact IBM Software Support.

Module: BNDA043

BND2733E Unable to connect to subsystem *ssss*.
RC=*xxxxxxxx*, REASON=*yyyyyyyy*.

Explanation: DB2 Bind Manager is unable to determine which version of DB2 is in use. In the message text, *ssss* is the subsystem ID; *xxxxxxxx* and *yyyyyyyy* are internal diagnostic codes for use by IBM Software Support.

Programmer response: Save all output. Contact IBM Software Support.

Module: BNDA043

BND2734S Unicode conversion failed for *ddddddd*,
symptoms=*xxxxxxxx/yyyyyyyy*.

Explanation: Unicode conversion was unsuccessful while processing a DBRM or package. In the message text, *ddddddd* is the name of the DBRM, *xxxxxxxx* is the return code from Unicode System Services, and *yyyyyyyy* is the reason code.

Programmer response: Inspect the DBRM to determine the correct character set ID. Make sure that Unicode System Services on your system is configured to convert the character set correctly.

Module: BNDA043

BND2799S Invalid call to message routine,
symptom=*nnn*.

Explanation: An internal processing error has been detected. In the message text, *nnn* is an internal diagnostic code for use by IBM Software Support.

Programmer response: Save all output. Contact IBM Software Support.

Module: BNDA043

BND2909I MEMBER *member_name* is not found in
DDNAME DBRMIN

Explanation: The DBRM name specified in the BIND statement was not found in the ddname DBRMIN. The BIND statement is passed to BINDOUT, but if the same library is used for the BIND it will fail also.

Programmer response: Verify that DBRM name specified in the BIND command exists in the DBRM library specified for ddname DBRMIN.

Module: BNDAVB3

BND2917I SYSPACKAGE ENTRY=*dbrmname* WITH
VERSION *version* NOT FOUND

Explanation: The new DBRM specified a version that did not exist in SYSIBM.SYSPACKAGE. The bind must be processed.

Programmer response: No action is required.

Module: BNDAVB3

BND2918I SYSPACKAGE ENTRY=*dbrmname* WITH
VERSION *version* HAD CONTOKEN
contoken_value AND MATCHES NEW
DBRM

Explanation: The new DBRM specified a version that exists in SYSIBM.SYSPACKAGE, and the contoken value matches. The bind will not be processed.

Programmer response: No action is required.

Module: BNDAVB3

BND2919I SYSPACKAGE ENTRY=*dbrm_name*
WITH VERSION *version* HAD
CONTOKEN *contoken_value* AND DOES
NOT MATCH NEW DBRM

Explanation: The new DBRM specified a version that exists in SYSIBM.SYSPACKAGE, and the CONTOKEN value does not match. The bind must be processed.

Programmer response: No action is required.

Module: BNDAVB3

BND2935E There is a DBRMLIB process failure.
The DDNAME *ddname* is missing.

Explanation: BNDAVB3 requires the missing ddname to process. Since the ddname is missing, the return code is set to 16 and processing terminates. The minimum required ddnames for BNDAVB3 are BINDIN, BINDOUT, and DBRMIN. SYSPRINT is not required, but it is the only location for any error messages.

Programmer response: Provide the missing ddname and perform the BNDAVB3 procedure again.

Module: BNDAVB3

BND2940I The BINDIN control cards do not
specify EXPLAIN(YES).

Explanation: The bind statement does not specify EXPLAIN or specifies EXPLAIN(NO). Processing continues but no access path analysis can be done for this bind of this DBRM.

Programmer response: If EXPLAIN(NO) was specified, change the BIND commands to specify EXPLAIN(YES).

Module: BNDAVB3

BND3100S Module *module_name* could not be
loaded, reason=*diagnostic_code*

Explanation: A required load module could not be accessed. In the message text, *module_name* is the name of the missing module.

BND3101E • BND310E

| **Programmer response:** Modify your system or TSO session configuration so that the required module is available via STEPLIB, link list, or another method.

| **Module:** BNDA075

| **BND3101E** **SYSUT1 must be allocated in the BNDC002**

| **Explanation:** The SYSUT1 work file has not been allocated.

| **Programmer response:** Verify that you are using the most current version of CLIST BNDC002 and that the CLIST has not been incorrectly modified.

| **Module:** BNDA075

| **BND3102E** **Enter one of the listed options**

| **Explanation:** An invalid option was specified.

| **Programmer response:** Specify a valid option on the panel.

| **Module:** BNDA075

| **BND3103S** **SQL ERROR** *type_of_SQL_request,*
type_of_SQL_request, **sqlcode=code_number**

| **Explanation:** An SQL statement failed unexpectedly.

| **Programmer response:** Record the information in the message text, and contact IBM Software Support.

| **Module:** BNDA075

| **BND3104E** **DBRM generation failed,**
reason=code_number

| **Explanation:** DB2 Bind Manager could not retrieve DBRM information from the catalog. The diagnostic return code indicates the failing code path.

| **Programmer response:** Record the information in the message text, and contact IBM Software Support.

| **Module:** BNDA075

| **BND3105E** **DBRM generation failed,**
reason=code_number

| **Explanation:** DB2 Bind Manager could not retrieve DBRM information from the catalog. The diagnostic return code indicates the failing code path.

| **Programmer response:** Record the information in the message text, and contact IBM Software Support.

| **Module:** BNDA075

| **BND3106S** **Unable to connect to subsystem**
subsystem_ID, **return code=code_number,**
reason=code_number

| **Explanation:** DB2 Bind Manager could not access the specified DB2 subsystem. The return code and reason code are from the call attachment facility.

| **Programmer response:** Ensure that the subsystem name is correct and that the subsystem is running and available. If the problem persists, analyze the the return and reason codes from the call attachment facility to determine why the subsystem is inaccessible. For more information, see the *DB2 Application Programming and SQL Guide* in the IBM Product Information Centers.

| **Module:** BNDA075

| **BND3107I** **There are no catalog entries matching your criteria**

| **Explanation:** The specified plan or package does not exist in the subsystem catalog.

| **Programmer response:** Ensure that the subsystem name and requested plan or package names are specified correctly and that the plan or package actually exists in the indicated subsystem.

| **Module:** BNDA075

| **BND3108E** **Only one entry may be processed at a time**

| **Explanation:** Multiple line commands were entered on a panel that can process only one entry at a time.

| **Programmer response:** Select only one line command, and press Enter.

| **Module:** BNDA075

| **BND3109E** *Data_set_name* **could not be allocated.**
Reason: code_number

| **Explanation:** The specified data set could not be allocated. The return code was issued by TSO ALLOCATE.

| **Programmer response:** Ensure that the data set name is spelled correctly, exists, and is not in use by another user.

| **Module:** BNDA075

| **BND310E** *dbrmlib* **IS NOT A VALID DBRM LIBRARY**

| **Explanation:** The specified load library is not valid. The load library must be a partitioned data set (PDS) with RECFM=U. One or more of these attributes is not correct.

| **Programmer response:** Specify the name of a valid load library.

Module: BNDC002

BND3110I DBRM *DBRM_name* stored in
data_set_name

Explanation: The requested DBRM was generated and saved. The data set name is the name of the DBRMLIB where the DBRM was saved.

Programmer response: No action is required.

Module: BNDA075

BND3111E *Data_set_name* could not be allocated.
Reason: *code_number*

Explanation: The specified data set could not be allocated. The return code was issued by TSO ALLOCATE.

Programmer response: Ensure that the data set name is spelled correctly, exists, and is not in use by another user.

Module: BNDA075

BND3112E Invalid DBRM library --
invalid_attribute_name

Explanation: A valid DBRMLIB is a partitioned data set with fixed-length 80-byte records. The specified data set lacks one or more of the required attributes. Possible values for *invalid_attribute_name* are "not partitioned," "LRECL", and "RECFM."

Programmer response: Ensure that the JCL contains a valid DBRMLIB in the DD statement and references a valid DBRM library with RECFM=FB. Resubmit the job.

Module: BNDA075

BND3114S Unable to connect to subsystem
subsystem_ID, return code=*code_number*,
reason=*code_number*

Explanation: DB2 Bind Manager could not access the specified DB2 subsystem. The return code and reason codes are from the call attachment facility.

Programmer response: Ensure that the subsystem name is correct and that the subsystem is running and available. If the problem persists, analyze the return codes and reason codes from the call attachment facility in the *DB2 Application Programming and SQL Guide* to determine why the subsystem is inaccessible.

Module: BNDA075

BND3115S *Line_command* is not a valid line
command; 'S' and '/' are the only
acceptable entries

Explanation: The specified line command is not valid.

Programmer response: Specify a valid line command.

| Valid line commands are S and /.

| Module: BNDA075

BND3117S Unable to disconnect from subsystem
subsystem_name, return code=*code_number*,
reason=*code_number*

| **Explanation:** DB2 Bind Manager could not disconnect from the specified DB2 subsystem. The return code and reason code are from the call attachment facility.

| **Programmer response:** Analyze the call attachment return and reason codes in the *DB2 Application Programming and SQL Guide* to determine why DB2 Bind Manager could not disconnect from the DB2 subsystem.

| Module: BNDA075

BND3118S *value_name value_name* failed,
RC=*code_number*

| **Explanation:** An ISPF service failed unexpectedly. In the message text, the value names are strings that identify the type of ISPF service request. The code number is the condition code that is returned by ISPF.

| **Programmer response:** Capture a screen shot that contains the entire message, and contact IBM Software Support.

| Module: BNDA075

BND3199E Logic error in BNDA075 message
routine, symptom=*number*

| **Explanation:** This message should not be issued in normal operation. It indicates that a message number that was generated internally by DB2 Bind Manager was invalid.

| **Programmer response:** Save all of the output, and contact IBM Software Support.

| Module: BNDA075

BND3201I EXECUTING BNDAVB4 *maintenance-level*

| **Explanation:** The BNDAVB4 program is running. The alphanumeric characters represent the maintenance level of the module.

| **Programmer response:** No action is required. If you report a problem, include the maintenance level.

| Module: BNDAVB4

BND3235E DBRM PROCESS FAILURE *ddname*
DDNAME MISSING

| **Explanation:** The JCL does not contain one or more of the following DD statements:

- | • DBRMIN DD

| • SOURCIN DD
 | • SOURCEOT DD
 | **Programmer response:** Add the missing statements,
 | and resubmit the job.
 | **Module:** BNDAVB1

| **BND3301S** *ddname* DD STATEMENT MISSING

| **Explanation:** A DD statement was not supplied for
 | *ddname*.
 | **Programmer response:** Add the missing DD statement
 | to the JCL and resubmit the job.
 | **Module:** BNDA055

| **BND3302S** UNABLE TO OPEN *ddname*,
 | REASON=*reason code*

| **Explanation:** The OPEN operation failed for the data
 | set that is associated with *ddname*. See *Enterprise PL/I for*
 | *z/OS Messages and Codes* for an explanation of the
 | reason code.
 | **Programmer response:** Correct the condition that is
 | preventing a successful OPEN operation and resubmit
 | the job.
 | **Module:** BNDA055

| **BND3303S** MODULE *module* COULD NOT BE
 | LOADED, REASON=*reason code*.

| **Explanation:** The FETCH operation failed for load
 | module *module*. See *Enterprise PL/I for z/OS Messages and*
 | *Codes* for an explanation of the reason code.
 | **Programmer response:** A common cause of this error
 | is a missing STEPLIB in the JCL. Correct the condition
 | and resubmit the job.
 | **Module:** BNDA055

| **BND3304E** NO SUBSYSTEM NAME HAS BEEN
 | SPECIFIED

| **Explanation:** A DB2 subsystem name was not
 | specified.
 | **Programmer response:** Supply an SSID statement with
 | a valid subsystem name and resubmit the job.
 | **Module:** BNDA055

| **BND3305E** INVALID STATEMENT

| **Explanation:** The statement that was entered is not
 | recognized or has incorrect syntax.
 | **Programmer response:** Correct or remove the invalid
 | statement and resubmit the job.
 | **Module:** BNDA055

| **BND3306E** SSID MAY ONLY BE SPECIFIED ONCE

| **Explanation:** Multiple SSID commands were entered.
 | Only one SSID command can be issued per run.
 | **Programmer response:** Remove the superfluous SSID
 | statement or statements and resubmit the job.
 | **Module:** BNDA055

| **BND3307E** MISSING OR INVALID SSID

| **Explanation:** The SSID statement was coded without
 | an operand, or the operand is invalid.
 | **Programmer response:** Correct the SSID statement and
 | resubmit the job.
 | **Module:** BNDA055

| **BND3308E** UNABLE TO CONNECT TO
 | SUBSYSTEM *subsystem*, RC=*return code*,
 | REASON=*reason code*

| **Explanation:** DB2 Bind Manager is unable to establish
 | a connection to DB2 subsystem *subsystem*. See *DB2*
 | *Application Programming and SQL Guide* for an
 | explanation of the return code and reason code.
 | **Programmer response:** Verify that the correct
 | subsystem name was coded on the SSID statement and
 | that the subsystem is available. Correct the condition
 | that is preventing the connection and resubmit the job.
 | **Module:** BNDA055

| **BND3309S** INVALID DATA IN
 | SYSIBM.SYSCOLUMNS,
 | SYMPTOM=*symptom*

| **Explanation:** The DB2 catalog contains invalid data.
 | The catalog might be corrupted.
 | **Programmer response:** Save all output and contact
 | IBM Software Support.
 | **Module:** BNDA055

| **BND3310E** UNABLE TO CONNECT TO
 | SUBSYSTEM *subsystem*, BIND
 | REQUIRED

| **Explanation:** DB2 Bind Manager is unable to establish
 | a connection to DB2 subsystem *subsystem* using plan
 | BNDA055. The plan has not been bound in this
 | subsystem, or the bind is out of date.
 | **Programmer response:** Verify that the correct
 | subsystem name was coded on the SSID statement and
 | that the bind for plan BNDA055 is current. If necessary,
 | bind the plan SBNDSAMP(*member*), where *member* is
 | BNDBIND, BNDBIND7, or BNDBIND8, depending on
 | the version of DB2 on your system. Resubmit the job.
 | **Module:** BNDA055

BND3311I CONNECTED TO SUBSYSTEM *ssid*

Explanation: DB2 Bind Manager has successfully connected to subsystem *ssid*.

Programmer response: None.

Module: BNDA055

BND3312I DISCONNECTED FROM *ssid*

Explanation: DB2 Bind Manager has successfully disconnected from subsystem *ssid*.

Programmer response: None.

Module: BNDA055

BND3315S DSNTIAR FAILURE, RC=*return code*

Explanation: DSNTIAR issued a non-zero condition code. The condition code that was issued by DSNTIAR is indicated by *return code*.

Programmer response: Verify that the version of DSNTIAR that is used is correct for the version of DB2 that is running in the specified subsystem. DSNTIAR is accessed through STEPLIB or through the system link list.

Module: BNDA055

BND3316W UNABLE TO PROCESS *dsname* (INVALID RECFM). BYPASSED.

Explanation: The data set that is referenced by *dsname* is not a load library. The data set does not have RECFM=U. The data set is not processed.

Programmer response: Specify a valid formatted load library name and resubmit the job.

Module: BNDA055

BND3317W UNABLE TO PROCESS *dsname* (NOT PARTITIONED). BYPASSED

Explanation: The data set that is referenced by *dsname* is not a PDS. The data set is not processed.

Programmer response: Specify a valid PDS load library name and resubmit the job.

Module: BNDA055

BND3318W UNABLE TO OPEN *dsname*. BYPASSED

Explanation: The data set that is referenced by *dsname* could not be opened for input. The data set is not processed.

Programmer response: Verify that the data set exists, is not empty, or is not in use by another job or user. Resubmit the job.

Module: BNDA055

BND3319E DESERV ERROR, UNABLE TO READ DIRECTORY FOR *dsname*

Explanation: The directory for data set *dsname* could not be accessed.

Programmer response: Verify that the data set exists, is not empty, and is not in use by another job or user. Resubmit the job.

Module: BNDA055

BND3337S UNSUPPORTED DB2 RELEASE - THIS VERSION OF DB2 IS NOT SUPPORTED

Explanation: This version of BNDA055 supports DB2 versions 10 and earlier. The version of DB2 that is being used with BNDA055 is not supported.

Programmer response: None.

Module: BNDA055

BND3341S RUN TIME ERROR OCCURRED IN ROUTINE

Explanation: An internal processing error occurred in BNDA055.

Programmer response: Save all output and contact IBM Software Support.

Module: BNDA055

BND3344E ALLOCATION FAILED, SYMPTOMS=*xxxx xxxx xxxxxxxx*

Explanation: BNDA055 was unable to dynamically allocate a requested library. In the message text, *xxxx* represents diagnostic information in hexadecimal format.

Programmer response: Contact IBM Software Support and provide them with the hexadecimal diagnostic information that was provided in the message text, and all other output that is associated with this error condition.

Module: BNDA055

BND3345E DEALLOCATION FAILED, SYMPTOMS=*xxxx*

Explanation: BNDA055 was unable to dynamically free a library as requested. In the message text, *xxxx* represents diagnostic information in hexadecimal format.

Programmer response: Contact IBM Software Support and provide them with the hexadecimal diagnostic information that was provided in the message text, and all other output that is associated with this error condition.

BND3395E • BND3399S

| **Module:** BNDA055

| BND3395E TOO MANY NCAL MEMBERS

| **Explanation:** The total number of members in all NCAL libraries exceeds 100000, which is the maximum number that BNDA055 can process in a single execution.

| **Programmer response:** Reduce the number of NCAL libraries by breaking the job into two smaller jobs.

| **Module:** BNDA055

| BND3396E NO LOADLIB STATEMENTS

| **Explanation:** No LOADLIB statements were coded; there are no NCAL members to check.

| **Programmer response:** Add one or more LOADLIB statements and resubmit the job.

| **Module:** BNDA055

| BND3397E TOO MANY LOADLIB STATEMENTS

| **Explanation:** More than 100 LOADLIB statements were coded. The maximum number of LOADLIB statements that BNDA055 can process in a single run is 100.

| **Programmer response:** Reduce the number of LOADLIB statements by breaking the job into two smaller jobs.

| **Module:** BNDA055

| BND3398E MISSING OR INVALID LOAD LIBRARY NAME

| **Explanation:** In a LOADLIB statement, the data set name is missing or has incorrect syntax.

| **Programmer response:** Correct the LOADLIB statement and resubmit the job.

| **Module:** BNDA055

| BND3399S ERROR IN BNDA055 - SYMPTOM=*nnn*

| **Explanation:** An internal processing error occurred in BNDA055.

| **Programmer response:** Save all output and contact IBM Software Support.

| **Module:** BNDA055

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502
Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and `ibm.com`[®] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol ([®] or [™]), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

You might need to refer to other sources of information when you are using DB2 Bind Manager.

This section lists the documentation that supports DB2 Bind Manager. Use the appropriate library for the version of DB2 that you are using.

Tip: To quickly locate a specific book, use the IBM Publications Center, which is located at www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi. After you enter your country information, click **Search for publications**, and enter the publication number (xxxx-xxxx) of the book that you want.

DB2 Version 9.1 for z/OS product information

- *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840
- *DB2 Version 9.1 for z/OS Application Programming and SQL Guide*, SC18-9841
- *DB2 Version 9.1 for z/OS Application Programming Guide and Reference for Java*, SC18-9842
- *DB2 Version 9.1 for z/OS Codes*, GC18-9843
- *DB2 Version 9.1 for z/OS Command Reference*, SC18-9844
- *DB2 Version 9.1 for z/OS Data Sharing: Planning and Administration*, SC18-9845
- *DB2 Version 9.1 for z/OS Diagnosis Guide and Reference*, LY37-3218
- *DB2 Version 9.1 for z/OS Diagnostic Quick Reference Card*, LY37-3219
- *DB2 Version 9.1 for z/OS Installation Guide*, GC18-9846
- *DB2 Version 9.1 for z/OS Introduction to DB2*, SC18-9847
- *DB2 Version 9.1 for z/OS Licensed Program Specifications*, GC18-9848
- *DB2 Version 9.1 for z/OS Messages*, GC18-9849
- *DB2 Version 9.1 for z/OS ODBC Guide and Reference*, SC18-9850
- *DB2 Version 9.1 Performance Monitoring and Tuning Guide*, SC18-9851
- *DB2 Version 9.1 for z/OS Program Directory*, GI10-8737
- *DB2 Version 9.1 for z/OS RACF Access Control Module Guide*, SC18-9852
- *DB2 Version 9.1 for z/OS Reference for Remote DRDA Requesters and Servers*, SC18-9853
- *DB2 Version 9.1 for z/OS Reference Summary*, SX26-3854
- *DB2 Version 9.1 for z/OS SQL Reference*, SC18-9854
- *DB2 Version 9.1 for z/OS Utility Guide and Reference*, SC18-9855
- *DB2 Version 9.1 for z/OS What's New?*, GC18-9856
- *DB2 Version 9.1 for z/OS XML Extender for z/OS Administration and Programming*, SC18-9857
- *DB2 Version 9.1 for z/OS XML Guide*, SC18-9858

DB2 Universal Database for z/OS, Version 8 product information

- *DB2 Administration Guide*, SC18-7413
- *DB2 Application Programming and SQL Guide*, SC18-7415
- *DB2 Application Programming Guide and Reference for Java*, SC18-7414
- *DB2 Command Reference*, SC18-7416
- *DB2 Data Sharing: Planning and Administration*, SC18-7417
- *DB2 Diagnosis Guide and Reference*, LY37-3201
- *DB2 Diagnostic Quick Reference Card*, LY37-3202
- *DB2 Installation Guide*, GC18-7418
- *DB2 Licensed Program Specifications*, GC18-7420
- *DB2 Messages and Codes*, GC18-7422
- *DB2 ODBC Guide and Reference*, SC18-7423
- *DB2 Reference Summary*, SX26-3853
- *DB2 Release Planning Guide*, SC18-7425
- *DB2 SQL Reference*, SC18-7426
- *DB2 Utility Guide and Reference*, SC18-7427
- *DB2 What's New?*, GC18-7428

- *DB2 XML Extender for z/OS Administration and Programming*, SC18-7431
- *Program Directory for IBM DB2 Universal Database for z/OS*, GI10-8566

DB2 Universal Database for z/OS, Version 7 product information

- *An Introduction to DB2 for OS/390*, SC26-9937
- *DB2 Administration Guide*, SC26-9931
- *DB2 Application Programming and SQL Guide*, SC26-9933
- *DB2 Application Programming Guide and Reference for Java*, SC26-9932
- *DB2 Command Reference*, SC26-9934
- *DB2 Data Sharing: Planning and Administration*, SC26-9935
- *DB2 Data Sharing Quick Reference Card*, SX26-3846
- *DB2 Diagnosis Guide and Reference*, LY37-3740
- *DB2 Diagnostic Quick Reference Card*, LY37-3741
- *DB2 Image, Audio, and Video Extenders Administration and Programming*, SC26-9947
- *DB2 Installation Guide*, GC26-9936
- *DB2 Licensed Program Specifications*, GC26-9938
- *DB2 Master Index*, SC26-9939
- *DB2 Messages and Codes*, GC26-9940
- *DB2 ODBC Guide and Reference*, SC26-9941
- *DB2 Reference for Remote DRDA Requesters and Servers*, SC26-9942
- *DB2 Reference Summary*, SX26-3847
- *DB2 Release Planning Guide*, SC26-9943
- *DB2 SQL Reference*, SC26-9944
- *DB2 Text Extender Administration and Programming*, SC26-9948
- *DB2 Utility Guide and Reference*, SC26-9945
- *DB2 What's New?* GC26-9946
- *DB2 XML Extender for OS/390 and z/OS Administration and Programming*, SC27-9949
- *DB2 Program Directory*, GI10-8182

Accessibility titles cited in this book

- *z/OS ISPF User's Guide, Volume 1*, SC34-4822
- *z/OS TSO/E Primer*, SA22-7787
- *z/OS TSO/E User's Guide*, SA22-7794

Index

A

accessibility 8
architecture 4

B

batch DBRM interpreter 64
benefits 2
Bind Avoidance
 features 73
 JCL requirements 73
 using 73
Bind Avoidance features
 BNDAVB1 program 74
 CCSID checking
 suppressing 81
 changing report destination 80
 checking for precompiler warnings 80
 coprocessor support 74
 DBRM generation 81
 DECLARE TABLE exception 81
bind plan 14
bind process 2
bind result 2
BNDA005 23
BNDA006 program 40, 66
 commands
 DBRM 37
 LMOD 37
 MEMLIST 37
 REMOTESYS 37
 SSID 37
 DBRM Cross Reference in Batch 36
 JCL statements 36
BNDA007 41
BNDA020 program 42
 commands 43
 JCL statements 43
BNDA024 program 45
 commands 46
 JCL statements 45
BNDA030 program
 commands 49
 JCL statements 48
 listing timestamps 48
BNDA036 program 59
 Checkpoint/Restart facility 58
 cleaning the catalog 50
 commands 52
 JCL statements 51
BNDA043 program
 commands 65
 JCL statements 64, 65
 output report 64
BNDA046 program 69
 cleaning the catalog 67
 commands 68
 JCL statements 67
BNDA055 program 61
 control statements 62

BNDA055 program (*continued*)
 input 62
 output 62
 running 61
 sample job 63
BNDAVB 18, 20
 copying to library 20
BNDAVB program 73
BNDAVB1 program 74
 JCL statements 75
BNDAVB3 78
BNDAVB3 program
 JCL statements 79
 using 78
BNDAVB4 program
 JCL statements 74
BNDBIND 14
 JCL sample 14
BNDC002 23
 modify 23
BROWSE function 29

C

catalog
 cleaning 42, 50, 67
 program
 listing timestamps 48
CCSID
 checking 81
 suppressing 81
changes
 technical 8
Checkpoint/Restart facility
 BNDA036 program 58
cleaning the catalog
 BNDA020 42
CLIST 20
CLIST BNDC002
 modify 23
codes 84
commands
 BNDA006 program 37
 BNDA020 program 43
 BNDA024 program 46
 BNDA030 program 49
 BNDA036 program 52
 BNDA043 program 65
 BNDA046 program 68
 wild cards 46, 49, 52, 68
 wildcards 37
configuration 13
 accessing from launchpad 21
 bind plan 14
 changing module name 18
 DB2 Bind Manager for DB2I 20
 JCL 18
 requirements 13
 steps 14
 summary 14

- conventions
 - highlighting v
- coprocessor support 2
 - configuring for Bind Avoidance 74
- cross-checking plans 26

D

- DB2 Admin Launchpad 21
- DB2 token
 - validate 1
- DBRM
 - analyze 29
 - BNDA006 program 36
 - BROWSE function 29
 - Cross Reference in Batch (BNDA006) 36
 - cross-reference report 36
 - GENERATE function 32
 - in batch 45
 - re-create 32, 45
 - rebuilder 32
 - package reconstruct 32
 - plan reconstruct example 32
 - reconstruct 32
- DBRM Checker 1, 4, 23, 84
 - requirements 13
- DBRM Checker main menu 26
- DBRM Cross Check panel 26
- DBRM Cross Reference in Batch (BNDA006) 36
- DBRM cross-reference
 - ISPF mode 23
 - load modules 13
 - requirements 13
 - return codes 84
 - using 23
- DBRM cross-reference for ISPF mode
 - setting up 23
- DBRM generation
 - activating for Bind Avoidance 81
 - activating for Bind Manager 2
- DBRM library selection 26
- DBRM load module cross-reference 1, 23
 - results 26
 - selection list 26
 - using 23
- DBRM results 26
- DBRM selection 26
- DBRM tools 23
- DBRMLIB 1
- DECLARE TABLE exception 2, 81
- DECLARE TABLE statements
 - ignoring 2
- default values 23
 - specifying 23
- diagnostic information
 - gathering 83
- disability 8
- documentation 7
- documentation changes 8
- DSNH 20
- dynamic mode
 - BNDA036 program 55

E

- embedded SQL
 - processing 2
- environment setup 23
- environment support 13
- error messages 84

F

- features 1
- FREE command 61

G

- GENERATE function 32

H

- high-level qualifier
 - specifying 23
- high-level qualifier for work files
 - specifying 23
- highlighting conventions v

I

- ignoring DECLARE TABLE statements 81
- index
 - creating
 - BNDA036 59
 - BNDA046 69
 - index availability
 - tables 34
 - index utility
 - determining available indexes for tables 34
 - table selection 34
- information updates
 - receiving 7
- installation 13
- ISPF dialog
 - starting 23
- ISPF mode
 - running 23
 - setting up environment for DBRM cross-reference 23
 - using 23
- IXU 34

J

- JCL
 - configure 18
 - customize 18
- JCL requirements
 - Bind Avoidance 73
- JCL sample 14
 - Bind Manager 18
 - static SQL load module 41
- JCL statements
 - BNDA006 program 36
 - BNDA020 program 43
 - BNDA024 program 45
 - BNDA030 program 48
 - BNDA036 program 51
 - BNDA043 program 65

JCL statements (*continued*)
 BNDA046 program 67
 BNDAVB1 program 75
 BNDAVB3 program 79
 BNDAVB4 program 74

job control statements
 BNDA006 36
 BNDA020 43
 BNDA024 45
 BNDA030 48
 BNDA036 51
 BNDA043 65
 BNDA046 67
 BNDAVB3 79

K

keyboard shortcuts 8

L

legal notices
 notices 131
 trademarks 132

LINKLST 20

load module
 comparisons 23
 static SQL report 23

load module library selection 26

LookAt vi

M

message retrieval tool
 LookAt vi

message severities 85

messages 86
 error 84
 methods for accessing vi

module name
 changing 18

MySupport 7

N

NOCC
 suppressing CCSID checking 81

notices 131

O

options
 checking for precompiler warnings 80
 coprocessor support 2, 74
 DBRM generation 2, 81
 DECLARE TABLE exception 2, 81
 suppressing CCSID checking (NOCC) 81

P

package
 cross-checking 28
 inspecting 28
 PKG 28

Package mode 28

package reconstruct 32

PKG 28

plan
 cross-checking 26
 inspecting 26

Plan mode 26

plan reconstruct
 example 32

precompiler
 detecting warnings 80
 functions 2

problems
 diagnostic information about 83

process flow 4

program directory 13

publications 7

R

re-creating DBRM in batch 45

report
 changing destination 80

requirements 13
 configuration 13, 14

return codes 84
 DBRM Checker 84
 DBRM cross-reference 84

running
 DB2 Bind Manager 13
 DB2 Bind Manager from DB2I 20

S

sample report
 BNDA006 sample output report
 BNDA006 sample report 40, 66

SBDNSAMP 14

SBDNSAMP library 14

SBNDCLST 23

SBNDLOAD 23

SBNDPEND 23

scenarios
 DB2 Bind Manager and DB2 Path Checker 6
 DB2 Path Checker 6
 using DB2 Bind Manager and DB2 Path Checker 6

screen readers and magnifiers 8

settings 23

SMF generation 20

SMF record format 79

SMF records
 generating 73, 79

SQL analysis tools 23

SQL statements
 validate 2

static SQL
 load module
 JCL sample 41
 report 41
 processing 2

summary of changes 8

support 13
 required information 83

SYSOUT listing
 listing 26
 verifying statements 26

T

- table
 - index availability 34
- table selection index utility 34
- technical changes 8
- timestamps
 - listing 48
- trademarks 132

U

- updates 7
- using BNDAVB3 78

W

- wild cards
 - BNDA024 program 46
 - BNDA030 program 49
 - BNDA036 program 52
 - BNDA046 program 68
- wildcards
 - BNDA006 program 37



Product Number: 5655-E43

Printed in USA

SC19-2751-04

