



DB2 Server for VSE & VM

Operation

Version 7 Release 5



DB2 Server for VSE & VM

Operation

Version 7 Release 5

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 217.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1987, 2007. All rights reserved.

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

Contents

About This Manual	v
Who Should Use This Manual	v
Conventions Used in This Manual	v
Components of the Relational Database Management System	v
Organization of this Manual.	vii
Syntax Notation Conventions	viii
SQL Reserved Words	xii
Related Information	xii

Summary of Changes	xv
Summary of Changes for DB2 Version 7 Release 5	xv
Enhancements, New Functions, and New Capabilities	xv

Chapter 1. Starting and Stopping the Application Server	1
About the Product	1
About Operating the Application Server	1
Starting the Application Server in a VSE Environment	1
Starting the Application Server in a VM Environment	2
Specifying Initialization Parameters.	3
Viewing or Changing the Initialization Parameters.	4
Obtaining More Information	4
Putting Initialization Parameters in VSE Source Members or in VM CMS Files	5
Overriding Parameters in a Source Member	6
Overriding a CMS Parameter File	7
Using Multiple User Mode and Single User Mode.	8
Starting the Application Server in Multiple User Mode.	8
Starting the Application Server in Single User Mode.	8
Specifying Parameters for User Programs.	9
Specifying CMS FILEDEF Commands for the Application Server	10
Overriding the Block Size.	10
Specifying an Alternative Tape Drive Address	11
Overriding CMS FILEDEF Commands	11
Disconnecting the Application Server	11
Stopping the Application Server	11

Chapter 2. Managing Online Support	17
Starting VSE Guest Sharing Support	17
VSE Guest Sharing Operator Responsibilities	17
Starting the Online Support	18
Displaying the Status of the Online Support	20
Stopping the Online Support	29
Password Implications on Online Resource Adapter Termination	32
Adding Online Connections Support	33
Removing Online Connections Support	34
Changing Online Default Support	36
Using Syntax Rules for DB2 Server for VSE CICS Transactions	37

Managing DRDA2 Task-Related User Exits	38
Setting Parameters Required to Autostart DRDA TRUE	39
Disabling DRDA TRUE Support Within CICS/VSE.	41

Chapter 3. Operating the Application Server **43**

Issuing DB2 Server for VSE & VM Operator Commands	43
From the VSE System Operator Console.	43
From the VM System Console	43
From the Application Server Operator Console	44
From a CICS Terminal.	44
From User Displays	45
DB2 Server for VSE Accounting	45
Using Alternate Accounting Files	45
Archiving the Database	46
Creating a User Archive	48
Archiving the Log	50
Archiving the Log to Tape in VM/ESA	50
Archiving the Log to Disk in VM/ESA	51
Forcing Users to End Their Work	51
Forcing DB2 Server for VSE Remote Users to End Their Work	53
VM/ESA FORCE Enhancements for CRR	54
Forcing DB2 Server for VM Inactive Users	55
Choosing a National Language	56
Monitoring Performance	57
COUNTER	58
RESET	61
RESET CRR LOGNAMES (VM only)	62
RESET HIGHSTOR	62
RESET INDOUBT	63
SET	63
SHOW ACTIVE	65
SHOW ADDRESS	69
SHOW BUFFERS	70
SHOW CONNECT	73
SHOW CRR LOGNAMES (VM Only).	80
SHOW DBCONFIG.	81
SHOW DBEXTENT.	84
SHOW DBSPACE	85
SHOW INDOUBT	87
SHOW INITPARM	89
SHOW INVALID	90
SHOW LOCK ACTIVE	93
SHOW LOCK DBSPACE	96
SHOW LOCK GRAPH	98
SHOW LOCK MATRIX	103
SHOW LOCK USER	106
SHOW LOCK WANTLOCK	108
SHOW LOG.	113
SHOW LOGHIST	114
SHOW POOL	118

SHOW PROC	122
SHOW PSERVER	125
SHOW SQLDBGEN	127
SHOW STORAGE	130
SHOW SYSTEM	133
SHOW USERS (Valid for VM only)	135
START PROC	137
START PSERVER	138
START TCPIP	140
STOP PROC	141
STOP PSERVER	143

Chapter 4. Recovering from System or Media Failures 147

General Recovery	147
Resolving DRDA 2 In-Doubt Logical Units of Work Useful CRR Commands (Valid for VM only)	148
Resolving CICS In-Doubt Logical Units of Work Recovery from Media Failure	152
	154

Chapter 5. Tracing the DB2 Server for VSE & VM System 155

DB2 Server for VSE & VM Tracing	155
Specifying CMS FILEDEF for a DB2 Server for VM Trace.	155
Specifying Job Control for a DB2 Server for VSE Trace	156
Tracing in Multiple User Mode	157
Tracing in Single User Mode	158
Tracing to a CMS File	158
Special Considerations for Multiple Activations of Tracing	159
Tracing the DB2 Server for VSE Resource Adapter	159
Choosing the Functions to Be Traced	159
Starting a Trace.	160
Stopping a Trace	165
Formatting DB2 Server for VM Trace Output	166
Formatting DB2 Server for VSE Trace Output.	168
Trace Formatter Control Statements	170
Format of DB2 Server for VSE & VM Trace Output	174
Isolating Problems.	177
ISQLMAP	177
ISQL Dumps in DB2 Server for VSE.	177
ISQL In-Core Trace	177

Appendix A. DB2 Server for VSE & VM Initialization Parameters. 179

Common Parameters	179
Common Tracing Parameters	191
DB2 Server for VSE Parameters	193
DB2 Server for VM Parameters	195
DB2 Server for VM Data Spaces Support Parameters	196

Appendix B. VMDSS Operator

Commands 199

COUNTER POOL	199
Syntax.	199
RESET POOL	202
Syntax.	202
SET POOL	203
Syntax.	203
SET SAVEINTV	205
Syntax.	205
SET TARGETWS	206
Syntax.	206
SHOW ACTIVE	206
Syntax.	206
SHOW BUFFERS	207
Syntax.	207
SHOW LOCK ACTIVE	208
Syntax.	208
SHOW LOCK GRAPH	209
Syntax.	209
SHOW POOL	209
Syntax.	209
SHOW SAVEINTV	211
Syntax.	211
SHOW TARGETWS	212
Syntax.	212
Internal Counters	213

Notices 217

Trademarks	219
----------------------	-----

Bibliography. 221

Index 225

Contacting IBM 235

Product information	235
-------------------------------	-----

About This Manual

Who Should Use This Manual

This manual describes how to start and stop the DB2 Server for VSE & VM database manager. It also describes how to monitor the application server after it is running, and how to restart it if it fails.

This manual is for people who operate the DB2 Server for VSE & VM Version 7 Release 1 database manager in either of the following environments:

- Virtual Machine/Enterprise System Architecture (VM/ESA[®]) system
- VM/ESA system with VSE running as a guest under VM, and accessing the application server on VM
- VSE/ESA[™].

You should be familiar with operating either a VSE system or VM virtual machines. Most of the things you must know to operate the DB2 Server for VSE & VM database manager are in this manual. For additional help, see your system administrator, or refer to the manuals listed in the bibliography.

This manual can also be useful if you require a reference guide for DB2 Server for VSE & VM initialization parameters or for operator commands.

Conventions Used in This Manual

The term *you* within the context of this manual means the operator of a DB2 Server for VSE & VM relational database management system (RDBMS); *user* and *end user* are the actual users of this system.

All references to CICS[®] in this manual are references to the CICS/VSE[®] product. In SQL/DS Version 3 Release 3 or later (including DB2 Server for VM Version 5 Release 1), you can connect to other IBM relational databases using the Distributed Relational Database Architecture[™] (DRDA[®]) implementation.

Components of the Relational Database Management System

Figure 1 on page vi depicts a typical configuration with one database and two users.

Figure 2 on page vii depicts a typical configuration with one database, one batch partition user, and a CICS[®] partition with several interactive users.

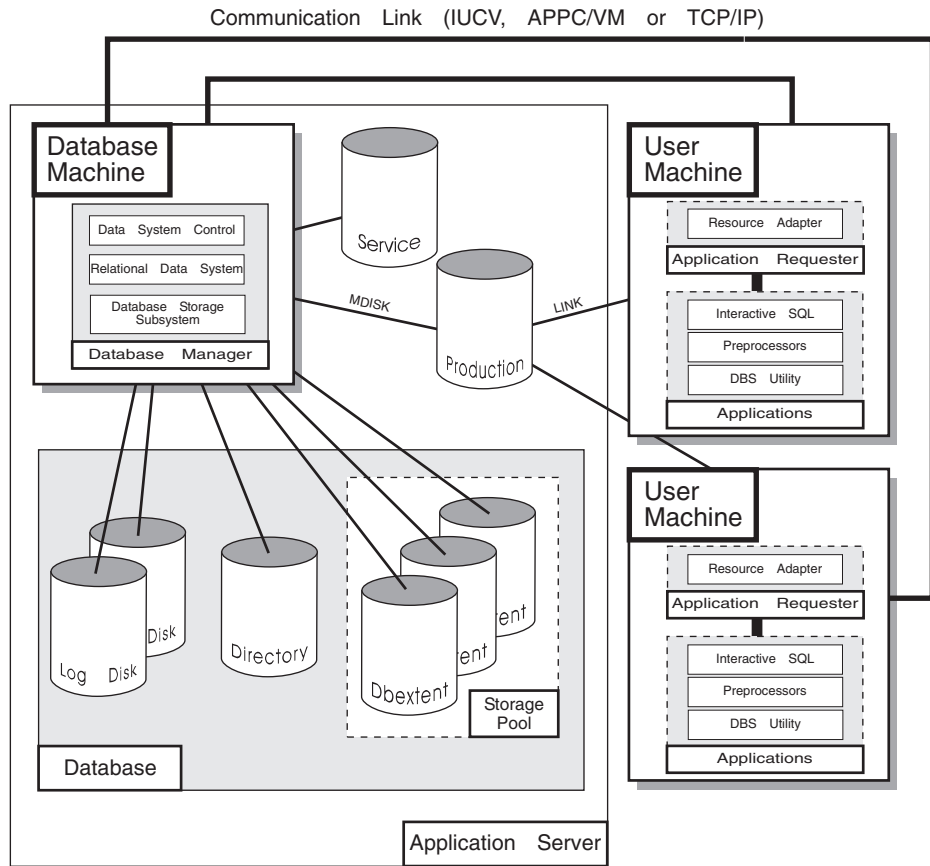


Figure 1. Basic Components of the RDBMS in VM/ESA

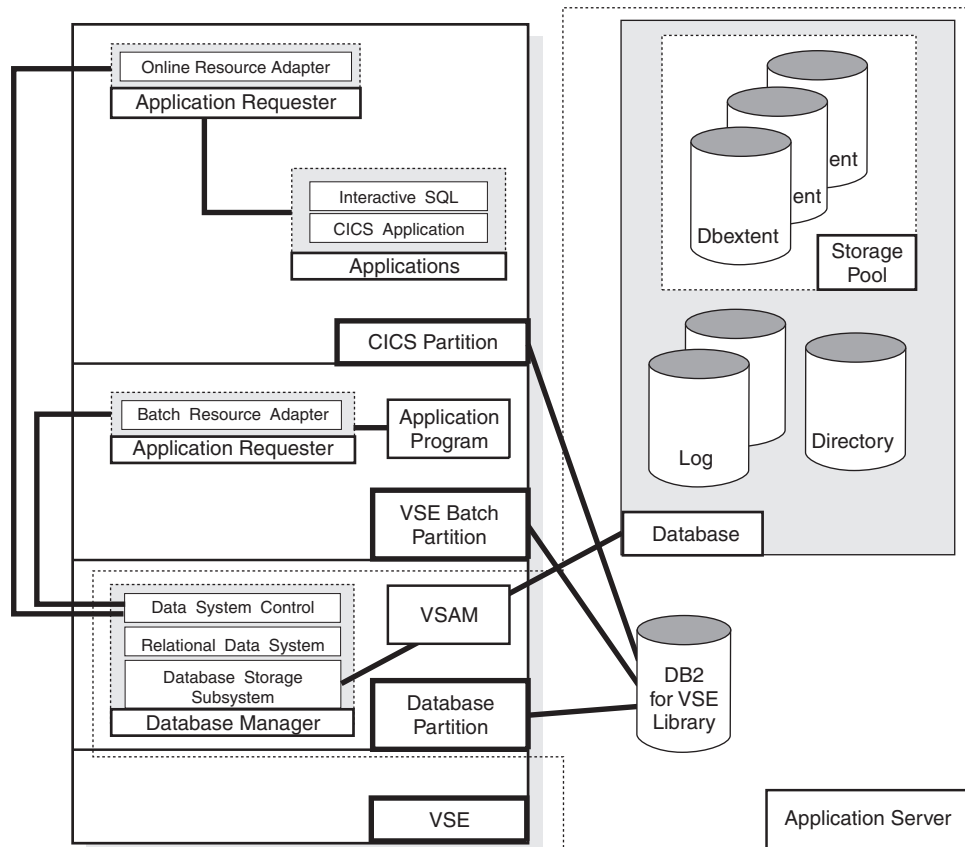


Figure 2. Basic Components of the RDBMS in VSE/ESA

The **database** is composed of :

- A collection of data contained in one or more *storage pools*, each of which in turn is composed of one or more *database extents (dbextents)*. A dbextent is a VM minidisk or a VSE VSAM cluster.
- A *directory* that identifies data locations in the storage pools. There is only one directory per database.
- A *log* that contains a record of operations performed on the database. A database can have either one or two logs.

The **database manager** is the program that provides access to the data in the database. In VM it is loaded into the database virtual machine from the production disk. In VSE it is loaded into the database partition from the DB2 Server for VSE library.

The **application server** is the facility that responds to requests for information from and updates to the database. It is composed of the database and the database manager.

The **application requester** is the facility that transforms a request from an application into a form suitable for communication with an application server.

Organization of this Manual

This manual is organized into the following chapters and sections:

Summary of Changes	Summarizes the technical and library changes made for the DB2 Server for VSE & VM Version 7 Release 1 database manager.
Chapter 1	Introduces the DB2 Server for VSE & VM database manager and explains how to start and stop the database management system.
Chapter 2	Explains how to: <ul style="list-style-type: none"> • start and stop the DB2 Server for VSE online support in a VSE environment • starting and stopping the DB2 Server for VSE online support for VSE Guest Sharing in a VM environment
Chapter 3	Shows how to do archiving, force users off the application server, and monitor the usage of the application server.
Chapter 4	Explains how to recover from system and media failures.
Chapter 5	Shows how to do a trace. If the database manager is the cause of a failure, you may have to trace its operation and format the output of that trace.
Appendix A	Lists and describes the application server initialization parameters.
Appendix B	Describes the operator commands included with VMDSS.
Bibliography	Lists manuals that contain information you may require when operating the database manager.

Syntax Notation Conventions

Throughout this manual, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right and from top to bottom, following the path of the line.

The ►— symbol indicates the beginning of a statement or command.

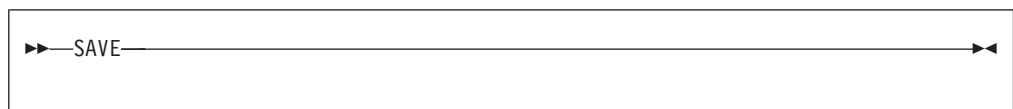
The —► symbol indicates that the statement syntax is continued on the next line.

The ►— symbol indicates that a statement is continued from the previous line.

The —► symbol indicates the end of a statement.

Diagrams of syntactical units that are not complete statements start with the ►— symbol and end with the —► symbol.

- Some SQL statements, Interactive SQL (ISQL) commands, or database services utility (DBS Utility) commands can stand alone. For example:



Others must be followed by one or more keywords or variables. For example:

A syntax diagram for the command `SET AUTOCOMMIT OFF`. It consists of a horizontal line with a double arrowhead on the left and a double arrowhead on the right. The text `SET AUTOCOMMIT OFF` is placed above the line.

- Keywords may have parameters associated with them which represent user-supplied names or values. These names or values can be specified as either constants or as user-defined variables called *host_variables* (*host_variables* can only be used in programs).

A syntax diagram for the command `DROP SYNONYM synonym`. It consists of a horizontal line with a double arrowhead on the left and a double arrowhead on the right. The text `DROP SYNONYM` is placed above the line, and *synonym* is placed below the line.

- Keywords appear in either uppercase (for example, `SAVE`) or mixed case (for example, `CHARacter`). All uppercase characters in keywords must be present; you can omit those in lowercase.
- Parameters appear in lowercase and in italics (for example, *synonym*).
- If such symbols as punctuation marks, parentheses, or arithmetic operators are shown, you must use them as indicated by the syntax diagram.
- All items (parameters and keywords) must be separated by one or more blanks.
- Required items appear on the same horizontal line (the main path). For example, the parameter *integer* is a required item in the following command:

A syntax diagram for the command `SHOW DBSPACE integer`. It consists of a horizontal line with a double arrowhead on the left and a double arrowhead on the right. The text `SHOW DBSPACE` is placed above the line, and *integer* is placed below the line.

This command might appear as:

```
SHOW DBSPACE 1
```

- Optional items appear below the main path. For example:

A syntax diagram for the command `CREATE UNIQUE INDEX`. It consists of a horizontal line with a double arrowhead on the left and a double arrowhead on the right. The text `CREATE` is placed above the line, `INDEX` is placed below the line, and `UNIQUE` is placed below the line between `CREATE` and `INDEX`.

This statement could appear as either:

```
CREATE INDEX
```

or

```
CREATE UNIQUE INDEX
```

- If you can choose from two or more items, they appear vertically in a stack. If you must choose one of the items, one item appears on the main path. For example:

A syntax diagram for the command `SHOW LOCK DBSPACE ALL integer`. It consists of a horizontal line with a double arrowhead on the left and a double arrowhead on the right. The text `SHOW LOCK DBSPACE` is placed above the line, `ALL` is placed below the line, and *integer* is placed below the line.

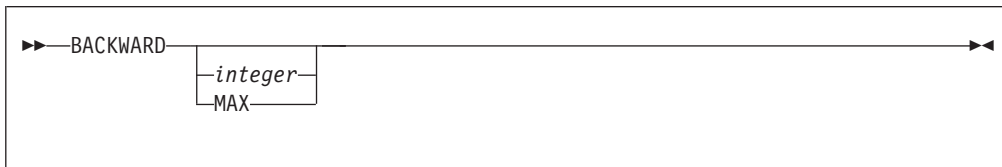
Here, the command could be either:

```
SHOW LOCK DBSPACE ALL
```

or

```
SHOW LOCK DBSPACE 1
```

If choosing one of the items is optional, the entire stack appears below the main path. For example:



Here, the command could be:

```
BACKWARD
```

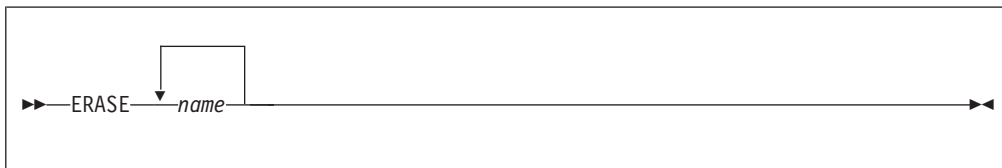
or

```
BACKWARD 2
```

or

```
BACKWARD MAX
```

- The repeat symbol indicates that an item can be repeated. For example:



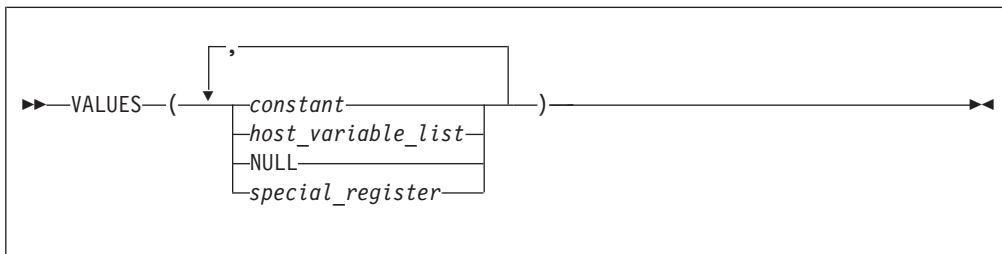
This statement could appear as:

```
ERASE NAME1
```

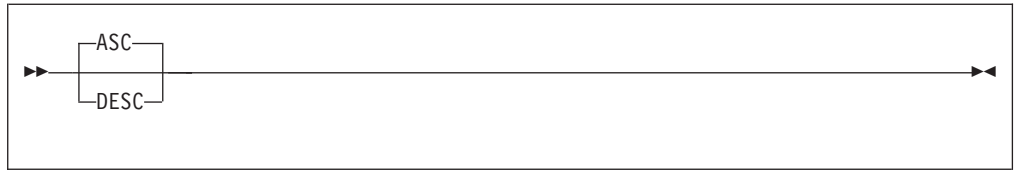
or

```
ERASE NAME1 NAME2
```

A repeat symbol above a stack indicates that you can make more than one choice from the stacked items, or repeat a choice. For example:

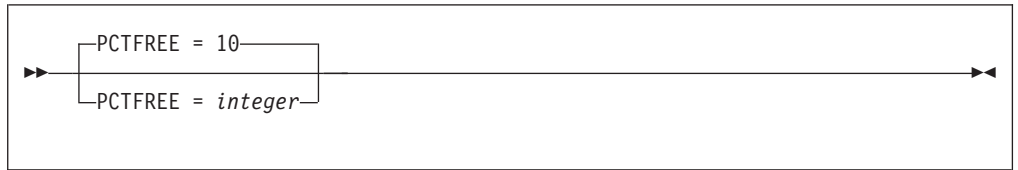


- If an item is above the main line, it represents a default, which means that it will be used if no other item is specified. In the following example, the ASC keyword appears above the line in a stack with DESC. If neither of these values is specified, the command would be processed with option ASC.

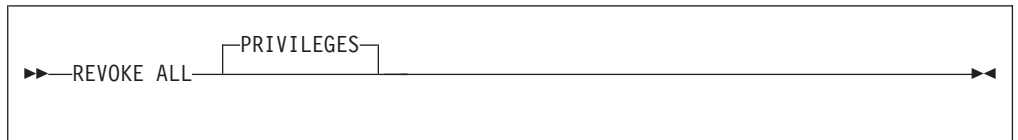


- When an optional keyword is followed on the same path by an optional default parameter, the default parameter is assumed if the keyword is not entered. However, if this keyword is entered, one of its associated optional parameters must also be specified.

In the following example, if you enter the optional keyword `PCTFREE =`, you also have to specify one of its associated optional parameters. If you do not enter `PCTFREE =`, the database manager will set it to the default value of 10.

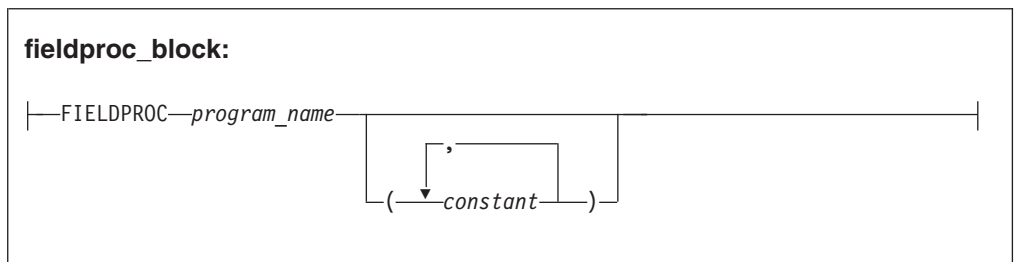
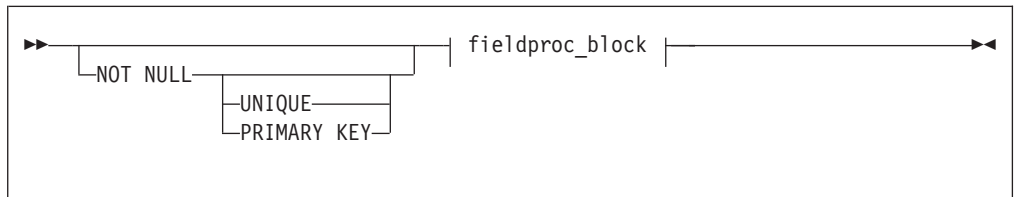


- Words that are only used for readability and have no effect on the execution of the statement are shown as a single uppercase default. For example:



Here, specifying either `REVOKE ALL` or `REVOKE ALL PRIVILEGES` means the same thing.

- Sometimes a single parameter represents a fragment of syntax that is expanded below. In the following example, **fieldproc_block** is such a fragment and it is expanded following the syntax diagram containing it.



SQL Reserved Words

The following words are reserved in the SQL language. They cannot be used in SQL statements except for their defined meaning in the SQL syntax or as host variables, preceded by a colon.

In particular, they cannot be used as names for tables, indexes, columns, views, or dbspaces unless they are enclosed in double quotation marks ("").

ACQUIRE	GRANT	RESOURCE
ADD	GRAPHIC	REVOKE
ALL	GROUP	ROLLBACK
ALTER		ROW
AND	HAVING	RUN
ANY		
AS	IDENTIFIED	SCHEDULE
ASC	IN	SELECT
AVG	INDEX	SET
	INSERT	SHARE
BETWEEN	INTO	SOME
BY	IS	STATISTICS
		STORPOOL
CALL	LIKE	SUM
CHAR	LOCK	SYNONYM
CHARACTER	LONG	
COLUMN		TABLE
COMMENT	MAX	TO
COMMIT	MIN	
CONCAT	MODE	UNION
CONNECT		UNIQUE
COUNT	NAMED	UPDATE
CREATE	NHEADER	USER
CURRENT	NOT	
	NULL	VALUES
DBA		VIEW
DBSPACE	OF	
DELETE	ON	WHERE
DESC	OPTION	WITH
DISTINCT	OR	WORK
DOUBLE	ORDER	
DROP		
	PACKAGE	
EXCLUSIVE	PAGE	
EXECUTE	PAGES	
EXISTS	PCTFREE	
EXPLAIN	PCTINDEX	
	PRIVATE	
FIELDPROC	PRIVILEGES	
FOR	PROGRAM	
FROM	PUBLIC	

Related Information

This manual neither describes how to select parameters that improve the performance of the DB2 Server for VSE & VM database manager, nor how to decide the best recovery procedures for your installation. That information is in the *DB2 Server for VM System Administration*, and *DB2 Server for VSE System Administration*.

After you (or someone at your installation) have made those decisions, you can use this manual to implement them. For example, after you have chosen parameters to improve the performance of the database manager, you can use this manual to find out how to specify the parameters correctly.

For information related to the DRDA environment, see the various manuals in the Distributed Relational Database Architecture Library, listed in “Bibliography” on page 221.

Summary of Changes

This is a summary of the technical changes to the DB2 Server for VSE & VM database management system for this edition of the book. Several manuals are affected by some or all of the changes discussed here. For your convenience, the changes made in this edition are identified in the text by a vertical bar (|) in the left margin. This edition may also include minor corrections and editorial changes that are not identified.

This summary does not list incompatibilities between releases of the DB2 Server for VSE & VM product; see either the *DB2 Server for VSE & VM SQL Reference*, *DB2 Server for VM System Administration*, or the *DB2 Server for VSE System Administration* manuals for a discussion of incompatibilities.

Summary of Changes for DB2 Version 7 Release 5

Version 7 Release 5 of the DB2 Server for VSE & VM database management system is intended to run on the Z/VM Version 5 Release 2 or later environment and on the Z/VSE[®] Version 3 Release 1 or later environment.

Enhancements, New Functions, and New Capabilities

The following have been added to DB2 Version 7 Release 5:

Explain Option on DBSU REBIND PACKAGE Command

This new functionality allows the EXPLAIN(YES/NO) option on REBIND PACKAGE command. If EXPLAIN(YES) is issued, then all four update tables (structure, plan, cost, reference) will be updated. If EXPLAIN(NO) is issued, then none of the four update tables will be updated.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VSE & VM Database Services Utility*
- *DB2 Server for VSE & VM Performance Tuning Handbook*
- *DB2 Server for VSE & VM Quick Reference*
- *DB2 Server for VSE & VM SQL Reference*

For Fetch only

This new functionality accepts the "FOR FETCH ONLY" clause after a cursor select statement. It causes a cursor to become read-only (no UPDATES or DELETES are permitted using this cursor). If a read-only cursor is referenced in an UPDATE or DELETE statement, SQLCODE -510 will be issued and the statement is not processed. In addition, under the SBLOCK preprocessor option, "FOR FETCH ONLY" forces blocking to be used on the read-only cursor regardless of whether there is a COMMIT. If there is no "FOR FETCH ONLY" clause, under SBLOCK, blocking would only be done if a COMMIT was absent.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VM Messages and Codes*
- *DB2 Server for VSE & VM Application Programming*
- *DB2 Server for VSE & VM Performance Tuning Handbook*
- *DB2 Server for VSE & VM Quick Reference*

- *DB2 Server for VSE & VM SQL Reference*

Application Message Formatter

This functionality provides an Application Programming Interface (API) that retrieves the descriptive text for an SQLCODE, given an SQLCA input parameter. The API will be available for Assembly, COBOL, C, PL/I and FORTRAN.

In DB2 for VM and DB2 for VSE Online, the user may specify the language of the returned text. The languages supported by DB2 for VSE/VM are American English (AMENG), uppercase English (UCENG), German (GER), French (FRANC) and Japanese (KANJI). VSE Batch does not support switching to another language. Therefore the default will be used regardless of the user's specification. The values of SQLCODE, SQLSTATE, SQLERRD1 and SQLERRD2 will be automatically appended to the returned text. The user may also specify to have the entire SQLCA included. If the SQLCODE could not be found in the repository, the entire SQLCA will be returned in the buffer.

If the SQLCA was set by another product (such as DB2 UBD), the descriptive text is retrieved if the SQLCODE exists in the DB2 for VM/VSE repositories. However, the token substitutions may not be correct.

For more information, see *DB2 Server for VSE & VM Application Programming*.

Convert buffer read/write to compiler macro

The DRDA code has over 100 small modules. Each call to an external module has a certain amount of overhead associated with it. Certain modules are called very frequently and this can add up to a significant amount of time. This functionality improves the performance by converting few modules to macros or internal procedures, to reduce this overhead.

Modify Build Tree Creation

This functionality modifies Build Tree creation used by DRDA parsing and generation. It is built in such a way that every code point that is used to search through the tree must be converted to a different format before the search can be done. If modified build tree was created with the converted point, then the code point would not have to be converted every time the tree must be searched. This improves the performance of the DRDA code path length with the minimal search.

Split code point search routines

When parsing a data stream within each parser action routine, a binary search is done to find the specific code point. Some action specific routines are quite large, so the binary search can be long. Splitting and spreading the code point evenly among other modules would reduce the overheads and improves the performance of the DRDA code path length.

DRDA Multi-Row Insert

Multi Row insert is a means of caching homogenous insert statements and sending them as a block to the server for processing. This reduces the overhead of sending a large number of singular inserts and receiving as many responses.

Buffering of homogenous inserts eliminates the need to send an SQL statement to the DB2 server every time an insert is made, thereby improving performance over DRDA.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VSE & VM Application Programming*

- *DB2 Server for VSE & VM Database Administration*
- *DB2 Server for VM System Administration*
- *DB2 Server for VSE & VM Performance Tuning Handbook*
- *DB2 Server for VSE & VM Quick Reference*
- *DB2 Server for VSE & VM SQL Reference*

Connection Pooling for DRDA TCP/IP in Online Resource Adapter

Connection pooling is a technique that allows multiple users to share a cached set of pre-established connections that provide access to a database. Establishing a connection between a user and a server takes a sizeable time. Users who have validated their entry to a database once need not establish a connection every time a request is submitted. Instead, they can use a pre-established connection from a pool of such connections and get their results much faster.

From the user's point of view, there is a considerable improvement in response time after this line item is implemented.

For more information, see the following documentation on DB2 Server for VSE & VM:

- *DB2 Server for VSE System Administration*
- *DB2 Server for VSE & VM Application Programming*
- *DB2 Server for VSE & VM Operation*
- *DB2 Server for VSE & VM Performance Tuning Handbook*

IBM DB2 Server for VSE, Client Edition

This feature allows the customer the flexibility to install and use only the client (run-time support) component of DB2 Server for VSE without the requirement to buy and install the server component during the installation process of DB2 server for VSE product. The client-only installation enables customers to reduce the total cost of ownership when they have their databases residing on a non-local platform (like VM, z/OS, LUW) and have a large number of their DB2 applications on VSE (like ISQL on CICS, DBSU on VSE, other online/batch applications on VSE).

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VSE System Administration*
- *DB2 Server for VSE Program Directory*

IBM DB2 Server for VM, Client Edition

This feature allows the customer the flexibility to install and use only the client (run-time support) component of DB2 Server for VM without the requirement to buy and install the server component during the installation process of DB2 server for VM product. The client-only installation enables our customers to reduce the total cost of ownership when they have their databases residing on a non-local platform (like VM, z/OS, LUW) and have a large number of their DB2 applications on VM (like ISQL, DBSU, other user applications on VM).

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VM System Administration*
- *DB2 Server for VM Program Directory*

Handling Commit Responses from DB2 UDB Stored Procedures

This feature will allow DB2 Resource Manager on VSE/VM to accept and process results of a stored procedure running in a UDB server with a COMMIT statement in the stored procedure.

Currently, DB2 for VM/VSE client does not handle responses from 'COMMIT' statements coded in DB2 UDB stored procedures. Implementation of this feature will enable handling responses of COMMIT statements in DB2 UDB stored procedures and thus allow users to have COMMIT statements in their stored procedures, while using DB2 for VM/VSE client.

COMMIT statements, however, are not allowed in stored procedures on the DB2 Server for VM/VSE.

For more information, see *DB2 Server for VSE & VM Application Programming*.

Make on-line programs AMODE 31 RMODE ANY

This feature converts DB2 server for VSE online program which presently operate under 24 bit addressing mode from AMODE 24, to AMODE 31 RMODE ANY. Presently, all the online programs are loaded below 16M line. Implementation of this line item ensures that all the online program will be loaded above the 16M line, which results in more virtual storage below the line, which can be utilized by other applications.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VSE System Administration*
- *DB2 Server for VSE Program Directory*

Provide BIND File Support in VM and in VSE Batch Environments

This feature provides the facility of binding packages across servers. The process of binding is achieved by dividing the program preparation method into two steps. The first step does the precompilation of the embedded SQL programs with the prep parameter 'BIND'. Invocation of VSE/VM preprocessor creates a 'bindfile'. The bindfile can be bound against any DB2 server using VSE/VM binder. During this process, the access path is generated, SQL statements are verified, authorization checks are performed, and package on the target server is created. This line item eliminates the need of re-prepping the source code or porting of packages across DB2 servers.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 REXX SQL for VM/ESA Installation and Reference*
- *DB2 Server for VM Messages and Codes*
- *DB2 Server for VSE & VM Application Programming*
- *DB2 Server for VSE & VM Database Administration*
- *DB2 Server for VM Program Directory*
- *DB2 Server for VSE Program Directory*

Convert TCP/IP LE/C interface to EZASMI API

The feature of converting TCP/IP LE/C interface to EZASMI API intends to replace the current LE/C interface and implement the EZA Assembler Interface (EZASMI) to enhance performance in DB2 Client/Server for VSE over DRDA. Currently, either LE/C interface or CSI Assembler Interface is used for TCP/IP functions. The EZASMI interface makes the code all Assembler.

For more information, see *DB2 Server for VSE Program Directory*

Chapter 1. Starting and Stopping the Application Server

This chapter introduces the application server, shows how to specify parameters when you are starting it, explains how to disconnect the DB2 Server for VM operator console (leaving the application server running), and describes how to stop the application server.

About the Product

The DB2 Server for VSE & VM product is a relational database management system (RDBMS) designed for both end users and data processing professionals. It is a flexible database management system that can accommodate many different application requirements. For example:

- Users can store, update, and retrieve information using a display.
- Users can store, update, and retrieve information from an application server.
- Users can restrict or limit access to information stored in a database.
- Users can maintain data integrity, even when an application program or system fails.

About Operating the Application Server

The application server runs in a partition of your VSE system and in a virtual machine of your VM system. To operate the application server, you must be able to start and stop it. If either your VSE installation has CICS, or your VM installation has VSE Guest Sharing and CICS, you must also be able to start and stop the DB2 Server for VSE & VM online support. The online support allows communication through CICS. Parameters for the online support for CICS users are explained in Chapter 2, “Managing Online Support,” on page 17.

When starting the application server in a VM environment, you can specify parameters that control the way the application server runs. These parameters control the number of users supported, the amount of virtual storage used, and so forth. This manual shows how to specify the parameters. Appendix A, “DB2 Server for VSE & VM Initialization Parameters,” on page 179 and the *DB2 Server for VM System Administration*, and *DB2 Server for VSE System Administration* describe the meaning of the parameters, and when they should be adjusted.

To view or change the initialization parameters once the application server has started, see “Viewing or Changing the Initialization Parameters” on page 4.

Starting the Application Server in a VSE Environment

You can start the application server in a VSE system just as you would start any batch job:

- By a set of job control statements entered with SYSIN (or SYSRDR)
- By job control commands entered from the operator console
- By cataloged procedures that you create.

The following is one example of job control (JCL) used to start the application server:

```
// JOB START SQL
// EXEC PROC=ARIS75DB
// EXEC PROC=ARIS75PL
// EXEC ARISQLDS,SIZE=AUTO
/*
/ &
```

For procedure ARIS75DB, substitute your own procedure that contains DLBL statements for your own database. (ARIS75DB identifies a starter database that is usually generated when you install this product.)

Procedure ARIS75PL contains the LIBDEF statements that refer to the DB2 Server for VSE libraries used in a production environment. Procedure ARIS75PL is created when you install this product.

Both procedures ARIS75DB and ARIS75PL are in other examples in this manual. If you are not sure which procedures you should specify, contact the person who installed this product.

When starting the application server, you can specify initialization parameters that determine how the application server operates. If you do not specify parameters, defaults are used. The previous example shows job control used to start the application server without specifying initialization parameters. Initialization parameters are usually specified at startup.

Starting the Application Server in a VM Environment

To start the application server in a VM/ESA system:

1. Log on to an application server. An application server is a virtual machine that has access to DB2 Server for VM code and a database.
2. Invoke the SQLSTART EXEC.

The SQLSTART EXEC is on the production disk, and the application server has access to it. This is the format of the SQLSTART EXEC:

```
►►—SQLSTART—Dbname—(server_name)———►
                        |dcssID(id)| |AMODE(nn)| |PARM(parameters)|
```

You must type the parentheses in the parameters. The parameters for the SQLSTART EXEC are:

Dbname(*server-name*)

is required. It identifies the database that the application server is to access. For example, to start the application server to access database SQLDBA, enter:

```
SQLSTART DBNAME(SQLDBA)
```

The DBNAME parameter must be the first or second parameter specified. For the DBNAME keyword, you can use any initial substring (such as DBNAM or DBNA).

dcssID(*id*)

is optional. Specify this parameter only if your installation has created saved segments for the DB2 Server for VM code and you want to use them. Otherwise, omit this parameter. For example, you can enter:

```
SQLSTART DBNAME(SQLDBA) DCSSID(SAVEDSEG)
```

If DCSSID is not specified, the *id* value from the *dbname* SQLDBN file on the production disk is used. If neither value is available, SQLDBA is used.

If you do not know what to specify for *id*, you can ask your VM system administrator. If specified, DCSSID must be the first or second parameter. You can specify ID instead of DCSSID for the keyword. No other abbreviation is valid. Refer to the *DB2 Server for VM System Administration* for more information about saved segments.

AMODE(*m*)

is optional. AMODE specifies the type of addressing mode (24-bit or 31-bit) the database manager runs in.

Refer to the *DB2 Server for VM System Administration* for more information about this parameter.

PARM(*parameters*)

is optional. You can use PARM to specify parameters that the DB2 Server for VM code uses. The application server uses parameters when you start it. These are called initialization or startup parameters. If you omit PARM, the application server uses defaults for its initialization parameters. See Appendix A, "DB2 Server for VSE & VM Initialization Parameters," on page 179 for a list of parameters and their defaults. If you specify PARM, you must put it last in the EXEC command. The PARM parameter cannot appear before DBNAME or DCSSID.

The initialization parameters in PARM define how the application server operates.

For example, if you intend to maintain the log and automatically archive the database, use the initialization parameter LOGMODE=A.

Because initialization parameters are optional, you can omit the PARM keyword. For example, you can start the application server to access the SQLDBA database without specifying initialization parameters:

```
SQLSTART DB(SQLDBA)
```

Note that you cannot run the application server in a conversational monitor system (CMS) batch machine.

Specifying Initialization Parameters

You specify initialization parameters after the PARM keyword in the SQLSTART EXEC statement. For example, if you want to use the following initialization parameters:

```
STARTUP=W  
LOGMODE=A  
DBNAME=SQLDBA
```

You can specify the parameters on the EXEC statement that invokes ARISQLDS in a VSE system like this:

```
// EXEC ARISQLDS,SIZE=AUTO,PARM='STARTUP=W,LOGMODE=A,DBNAME=SQLDBA'
```

The PARM keyword identifies a string of initialization parameters. Enclose the string of parameters in single quotation marks, as shown.

Note: The DBNAME you choose, for example 'SQLDBA' as shown above, must be defined in the DBNAME directory. Refer to the *DB2 Server for VSE System Administration* for information regarding the DBNAME directory.

You can specify the server name in DBNAME, specify the initialization parameters in PARM, and start the application server in a VM system like this:

```
SQLSTART DBNAME(SQLDBA) PARM(STARTUP=W,LOGMODE=A)
```

In these examples, commas separate the initialization parameters. You can also use blanks to separate the parameters, but do not put blanks within initialization parameters.

Viewing or Changing the Initialization Parameters

You can view the initialization parameters using the SHOW INITPARM command as follows:

```
▶—SHOW INITPARM—◀
```

For more information on SHOW INITPARM, see “SHOW INITPARM” on page 89.

You can change the settings of several initialization parameters without having to stop and restart the application server using the SET command. For more information on the SET command, see Appendix A, “DB2 Server for VSE & VM Initialization Parameters,” on page 179 and “SET” on page 63.

Obtaining More Information

For more information about initialization parameters, talk to your system administrator, or refer to Appendix A, “DB2 Server for VSE & VM Initialization Parameters,” on page 179.

If you receive error messages refer to the *DB2 Server for VM Messages and Codes*. For more detailed information about starting the application server, refer to the *DB2 Server for VM System Administration*, or the *DB2 Server for VSE System Administration*.

Putting Initialization Parameters in VSE Source Members or in VM CMS Files

When starting the application server, you often use the same set of initialization parameters. You can place the initialization parameters in either a VM CMS file or in a VSE A-type source member. You can use the CMS file instead of entering the parameters each time you start the application server, you merely refer to the file. Or, you can place the initialization parameters in a VSE A-type source member. Then, instead of entering the parameters each time you start the application server, you merely refer to the member.

The following is an example of the content of a parameter file in VM.

```
LOGMODE=A,NDIRBUF=20,  
NPAGBUF=20,  
DUMPTYPE=F,          COMMENT -- FULL VIRTUAL MACHINE DUMP  
NCSCANS=20
```

Figure 3. Example Parameter File

You can have as many of these parameter files as needed.

Notes:

1. In this example, use spaces in the place of commas when only one parameter appears on a line. Any text included on the same line, but following the space, will be interpreted as a comment.
2. When naming a parameter file, use any valid file name.
3. The file type must be SQLPARM and have a fixed record length of 80 bytes.

To refer to a file when running SQLSTART, put PARMID=file name in the PARM parameter. (PARMID is itself an initialization parameter.) For example, if the file name is WARMSTRT, and you want to access the SQLDBA application server, you can enter:

```
SQLSTART DB(SQLDBA) PARM(PARMID=WARMSTRT)
```

The application server reads the parameters from WARMSTRT SQLPARM (using * as the CMS file mode).

When placing initialization parameters in a CMS file, follow these rules:

- A blank following a parameter ends the processing of the line; therefore, do not put a blank between parameters. The application server ignores any characters that follow the blank. You can, however, use blanks to put comments in the file, as shown in Figure 3 on page 5 for the DUMPTYPE parameter.
- The file content must be in UPPERCASE.
- A comma at the end of a line is not needed, but can make the statement easier to read.
- You cannot put user parameters (those for the application program itself) in the file. See “Specifying Parameters for User Programs” on page 9 for more information. If the application server detects parameters other than its own initialization parameters, it issues error messages and stops processing.

The following is an example of a job you might use to catalog a source member in VSE.

```
// JOB CATALPRM
// EXEC LIBR
ACCESS SUBLIB=PRD2.SQL340
CATALOG PARMXMPL.A,REPLACE=YES
DSPLYDEV=B,NDIRBUF=20,SYSMODE=S,
PROGRAM=USERPROG,NPAGBUF=20,
DUMPTYPE=F, COMMENT - FULL PARTITION DUMP
NCSCANS=20
/*
/ &
```

Figure 4. Example Job to Catalog a Source Member

You can have as many source members as you need. The member must be an A-type source member. Each member specifies a different set of frequently used parameters.

To refer to the member when starting the application server, specify `PARMID=member_name`. (`PARMID` is itself an initialization parameter.) For example, if the name of the member is `PARMXMPL`, you can use:

```
// EXEC ARISQLDS,SIZE=AUTO,PARM='PARMID=PARMXMPL'
```

The application server reads the parameters from the `PARMXMPL` source member.

Follow these rules when placing initialization parameters in either a CMS file or in a source member:

- A blank following a parameter ends the processing of the line; therefore, do not put a blank between parameters. The application server ignores any characters that follow the blank. You can, however, use blanks to put comments in the file, as shown in Figure 4 on page 6 for the `DUMPTYPE` parameter.
- A comma at the end of a line is not needed, but can make the statement easier to read.
- You cannot put user parameters (those for the application program itself) in the file. See “Specifying Parameters for User Programs” on page 9 for more information. If the application server detects parameters other than its own initialization parameters, it issues error messages and stops processing.

Overriding Parameters in a Source Member

Even if you do not use exactly the same set of parameters every time you start the application server, it is still convenient to use a source member for parameters. You can override parameters in the member by specifying them again when you invoke the application server.

Usually, you would start the application server with the following job control:

```
// JOB START SQL
// EXEC PROC=ARIS75DB
// EXEC PROC=ARIS75PL
// EXEC ARISQLDS,SIZE=AUTO,PARM='PARMID=STARTUP'
/*
/ &
```

Now, suppose you want to start the application server, but want to use NLRBU=1500 instead of NLRBU=1000. You would do this:

```
// JOB START SQL
// EXEC PROC=ARIS75DB
// EXEC PROC=ARIS75PL
// EXEC ARISQLDS,SIZE=AUTO,PARM='PARMID=STARTUP,NLRBU=1500'
/*
/ &
```

Any parameter that you specify for the EXEC statement overrides any that occurs in the referenced member.

When all the values of the initialization parameters have been resolved, the application server displays the final values as routed by the DSPLYDEV initialization parameter.

Overriding a CMS Parameter File

Even if you do not use exactly the same set of parameters every time you start the application server, it is still convenient to use a parameter file. You can override parameters in the file by specifying them again when invoking SQLSTART. Suppose file STARTUP SQLPARM contains the following:

```
NCUSERS=10
ARCHPCT=90
LOGMODE=A
NLRBU=1000
```

Usually, you would start the application server to access database SQLDBA like this:

```
SQLSTART DB(SQLDBA) PARM(PARMID=STARTUP)
```

Now, suppose you want to start the application server, but want to use NLRBU=1500 instead of NLRBU=1000. You would enter:

```
SQLSTART DB(SQLDBA) PARM(PARMID=STARTUP,NLRBU=1500)
```

Any parameter that you specify for the command overrides any that occurs in the referenced CMS file.

Using Multiple User Mode and Single User Mode

The DB2 Server for VSE & VM application server can run in one of two modes:

- Multiple user mode (MUM)
- Single user mode (SUM).

Usually the DB2 Server for VSE system runs alone in its own partition. Users can use the application server from their own partitions. The system provides for the communications between the DB2 Server for VSE partition and user partitions. This standard mode of operation is multiple user mode. For information on starting the application server in multiple user mode, refer to “Starting the Application Server in a VSE Environment” on page 1, and “Starting the Application Server in a VM Environment” on page 2.

Starting the Application Server in Multiple User Mode

VM users can access the application server from their own virtual machines. The DB2 Server for VM system provides for the communications between the application server and the application requester. The standard mode of operation is multiple user mode. In the VM/ESA system, remote users as well as local users can access the application server in multiple user mode.

When you enter the SQLSTART command and do not specify any parameters, the system is in multiple user mode.

Starting the Application Server in Single User Mode

For some applications, there is an advantage to running in single user mode. In VM, single user mode means that users can access the application server from their own virtual machines; in VSE it means that applications run in the same partition as the application server. The system handles the data requests for only that one application and other users and applications cannot access the application server.

In single user mode, VSE guests in a VM system cannot access a VM database, nor can they use the DRDA protocol.

To start the application server in single user mode, you must specify the initialization parameter `SYSMODE=S`. You must also specify the `PROGNAME` parameter. This parameter indicates to the application server the name of the application to run. The format for the parameter is `PROGNAME=name` where `name` is the name of the user program.

Note: DB2 Server for VM ignores the `PROTOCOL` parameter when running single user mode.

If your application is running in VSE and the program requires its own job control statements (such as `TLBL` or `DLBL` statements), you should include those statements in the job control to start the application server.

```
// JOB SINGLE
// EXEC PROC=ARIS75PL
// EXEC PROC=DBNAME01
// EXEC PGM=ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,PROGNAME=name'
/*
/&
```

Figure 5. Job Control to Start the Application Server in Single User Mode on VSE

In VM, when the application server is started, it branches to the program specified in the PROGNAME parameter without first giving CMS control.

If the program has any special requirements, such as CMS FILEDEFS, you must do them before invoking SQLSTART.

Note: If you do not specify single user mode, the application server defaults to multiple user mode.

Figure 6 shows the options for starting the application server in single user mode.

```
SQLSTART DBNAME(server_name) PARM(SYSMODE=S,PROGNAME=name)
```

Figure 6. Starting the Application Server in Single User Mode on VM

When the user program ends, the application server ends.

Specifying Parameters for User Programs

Some programs require input parameters similar to those that the application server requires. In multiple user mode, users run their own programs and they provide whatever parameters the programs require. In single user mode, you must specify the user parameters when starting the application server.

The following is an example of starting the application server to run program PATSCAN. To run program PATSCAN, you must specify the following parameters.

```
COPY,CONT,KEEP
```

The person who writes the program determines the required parameters and their format.

You specify the user parameters in PARM; they must follow the DB2 Server for VSE & VM parameters as shown. The slash (/) separates the DB2 Server for VSE & VM parameters from the user parameters.

You can specify, at most, 100 characters for the PARM parameter (including blanks and commas). If you specify many parameters, and user parameters as well, there might not be enough room on the EXEC statement or command. If there is not enough room, place all the DB2 Server for VSE parameters in a source member and just specify PARMID on the command line.

Note: You cannot put user parameters in a source member. In the above example, the DB2 Server for VSE parameter PROGNAME cannot be put in a source member because it is passing user parameters COPY, CONT, and KEEP to the program.

To start the application server to run the program in VSE, enter:

```
// JOB SINGLE
// EXEC PROC=IMAGE
// EXEC PROC=ARIS75PL
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,PROGNAME=PATSCAN/COPY,CONT,KEEP'
/*
/ &
```

To start the application server to run the program in VM, enter:

```
SQLSTART DB(IMAGE) PARM(SYSMODE=S,PROGNAME=PATSCAN/COPY,CONT,KEEP)
```

If there is not enough room in the CMS input area, place all the DB2 Server for VM parameters, except for PROGNAME, in a parameter file and specify PARMID and PROGNAME on the command line. Remember, you cannot put user parameters in that file.

Specifying CMS FILEDEF Commands for the Application Server

The SQLSTART EXEC issues these CMS FILEDEF commands for the application server:

```
FILEDEF ARIARCH TAP1 SL (BLKSIZE 4096 NOCHANGE PERM
FILEDEF ARITRAC TAP2 SL (BLKSIZE 4096 NOCHANGE PERM
FILEDEF ARILARC TAP3 SL (BLKSIZE 4096 NOCHANGE PERM
FILEDEF ARILALT TAP4 SL (BLKSIZE 4096 NOCHANGE PERM
```

The application server uses them for database archiving, tracing, and log archiving. Database archiving is described in “Archiving the Database” on page 46. Tracing is described in 161. Log archiving is described in “Archiving the Log” on page 50.

Overriding the Block Size

You can use the BLKSIZE option of the CMS FILEDEF command to override the block size for:

- Database archive file output (ARIARCH)
- Log archive file output (ARILARC, ARILALT).

Note: The FILEDEF command must be issued in CMS mode, prior to the startup of the application server. A 28 kilobyte block size is used for log archive and database archive file input; you cannot override this value.

You can specify any output block size from 4 kilobytes to 28 kilobytes in multiples of 4 kilobytes. You must do this before starting the application server by specifying a new BLKSIZE option that will override the ARIARCH, ARILARC, and ARILALT FILEDEF commands. For example, you can enter a CMS command similar to this:


```
FILEDEF ARIARCH TAP1 SL (BLKSIZE 28672 CHANGE PERM
```

In the above diagram, the block size is 28 kilobytes (28 672 bytes).

Specifying an Alternative Tape Drive Address

You can use the ALT option of the CMS FILEDEF command to specify an alternative tape drive virtual device number. You can use it, for example, if you do a log or database archive that uses multiple tape volumes. If you specify an alternative tape drive address by overriding the FILEDEF command, the archive switches automatically to the other tape drive when the end of the tape is reached.

To specify an alternative tape drive with the FILEDEF command for the database archive file, enter the following command:

```
FILEDEF ARIARCH TAP1 SL (ALT cuu
```

In the above example, cuu represents the virtual device address of the alternative tape drive. For more information, refer to the *VM/ESA: CMS Command Reference*.

Overriding CMS FILEDEF Commands

If you override the FILEDEF commands when running a program in single user mode, specify the PERM option on your FILEDEF. You can omit the PERM option if the program is in Assembler language. For all other languages, you must specify the PERM option.

The *DB2 Server for VM System Administration* describes reasons you may want to enter FILEDEF commands for the application server.

Disconnecting the Application Server

To sign off the application server without stopping the DB2 Server for VM machine, enter these commands:

```
#CP SET RUN ON  
#CP DISCONN
```

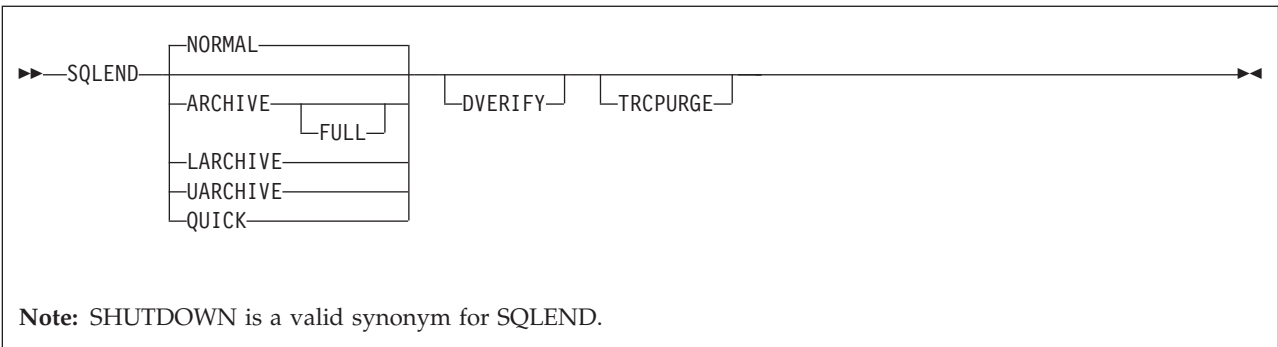
Stopping the Application Server

You can stop the application server when in multiple user mode by issuing the SQLEND operator command.

In single user mode, the application server stops after the application terminates, and displays informational messages. In all situations, the database manager closes the database:

- CMS files and trace files (if any are active).
- VSE accounting, and trace files (if active).

The syntax of the SQLEND command is as follows:



Keyword	Description
NORMAL	<p>indicates that new users cannot access the application server. When all current users are finished, the application server ends.</p> <p>Note: New users refers to users who are not currently connected to the application server. Whether active or not, both types of users are classified as existing or current users. (NORMAL is the default.)</p>
ARCHIVE	<p>specifies that new users cannot access the application server. When all current users are finished, the database manager archives the entire database and ends. You can specify ARCHIVE when the application server is running with LOGMODE set to A, Y, or L. If LOGMODE=L, the active log is archived if there is information in the log before archiving the database. If alternate logging is enabled, both the inactive log and the active log are archived if the inactive log was not archived previously.</p> <p>If LOGMODE=Y, the application server automatically switches to LOGMODE=A to create the database archive. To continue to run with LOGMODE=Y, you must do a COLDLOG to switch back to LOGMODE=Y from LOGMODE=A.</p> <p>FULL an archive taken with the FULL option can be used as the full backup associated with an incremental backup taken with the Data Restore feature.</p>
LARCHIVE	<p>indicates that new users cannot access the application server. When all current users are finished, the system archives the active log, which contains a record of changes made to the database, and ends. If alternate logging is enabled, both the inactive log and active log are archived if the inactive log was not archived previously. You can specify LARCHIVE only when LOGMODE=L. If you specify LARCHIVE when LOGMODE is not set to L, an error message is displayed.</p>
UARCHIVE	<p>specifies that new users cannot access the application server. When all current users are finished, the application server indicates in its log that a user archive will be done and the application server ends. Then you must archive the directory and dbextents using your own backup utilities and procedures. Do not archive the log with non-DB2 Server for VSE & VM facilities.</p> <p>You can specify UARCHIVE regardless of the LOGMODE setting. If LOGMODE=L, the database manager archives the active log before ending if there is information in the active log. If alternate</p>

logging is enabled, both the inactive log and active log are archived if the inactive log was not archived previously.

If LOGMODE=Y, the database manager automatically switches to LOGMODE=A to prepare for the user archive. To continue to run with LOGMODE=Y, you must do a COLDLOG to switch back to LOGMODE=Y from LOGMODE=A.

QUICK causes the database manager to end immediately. All users are immediately disconnected from the application server or application server partition.

DVERIFY causes the application server to verify its directory before ending. You can specify this keyword with any other SQLEND keyword or by itself. When SQLEND is specified without keywords, a normal shutdown is implied. When SQLEND is specified with QUICK, the application server ignores DVERIFY.

The application server displays a message if it finds an error in its directory. If there is an error, and you have specified ARCHIVE, the application server does not archive the database. If you specified ARCHIVE and LOGMODE to be set to L, the database manager does the log archive if there is information in the log, but does not do the database archive.

When you specify LARCHIVE, and there are discrepancies in the directory, the database manager does the log archive anyway, because the discrepancies are in the directory, not the log.

If you specified UARCHIVE and there is an error in the directory, do not archive the database. (A message reminds you not to do the archive.) If LOGMODE is set to L when you specify UARCHIVE, the database manager archives the log if there is information in the log. Refer to the *DB2 Server for VSE & VM Diagnosis Guide and Reference* for information on recovering from directory errors.

TRCPURGE specifies if you want to purge the contents of the trace buffer at DB2 Server for VSE & VM shutdown. An output trace file will not be created for the trace buffer in this case. The TRCPURGE parameter is ignored if there is no trace buffer in use when SQLEND is issued.

The following table summarizes archive activity at shutdown based on the SQLEND parameters and the LOGMODE setting. See “Archiving the Database” on page 46 for an explanation of logmodes N, Y, L, and A.

Table 1. Summary of Archives Done at Shutdown

SQLEND Parameter	Logmode			
	N	Y	L	A
NORMAL	Not applicable	None	None	None
QUICK	Not applicable	None	None	None
ARCHIVE	Not applicable	The database manager archives the database. ^{1, 2, 3}	The database manager does a log archive if there is information in the log, and then does a database archive. ^{1, 3}	The database manager archives the database. ^{1, 3}

Table 1. Summary of Archives Done at Shutdown (continued)

SQLEND Parameter	Logmode			
	N	Y	L	A
LARCHIVE	Not applicable	You cannot specify LARCHIVE when LOGMODE=Y. (An error message is displayed.)	The database manager does a log archive.	You cannot specify LARCHIVE when LOGMODE=A. (An error message is displayed.)
UARCHIVE	Not applicable	The operator archives the database after shutdown. ^{1, 2}	The database manager does a log archive if there is information in the log; then the operator archives the database after shutdown. ¹	The operator archives the database after shutdown. ¹

Notes:

- ¹ If you have specified DVERIFY, and DVERIFY is successful, the database manager does the archives shown in Table 1 on page 13. You should do user archives as appropriate. If DVERIFY fails, the database manager does all log archives indicated, but not the database archive. Any inconsistency in the database is recorded in the database archive, and a subsequent restore using that archive will fail. Do not do a user archive when DVERIFY fails.
- ² If LOGMODE is set to Y when you issue the SQLEND command, and you specify either ARCHIVE or UARCHIVE, LOGMODE is automatically reset to A.
- ³ If the FULL parameter is specified, the archive can be used as the full backup associated with an incremental backup taken with the Data Restore feature.

Because NORMAL, ARCHIVE, LARCHIVE, and UARCHIVE let users complete their work, a delay occurs before the application server ends. After entering SQLEND with one of these keywords, you can enter the SHOW USERS command to determine who is still using the application server. For more information about the SHOW USERS command, see "SHOW USERS (Valid for VM only)" on page 135.

When you specify ARCHIVE or LARCHIVE (or UARCHIVE with LOGMODE=L), the database manager writes information to tape, unless you are in a VM environment and you are taking a log archive to disk. After all database manager activity stops, if TAPEMGR=N, the database manager then requests that you mount the required tape volume(s) to store the archive. At the same time, it displays information that you can use to label the tape externally.

If TAPEMGR=N the database manager prompts you to mount and prepare the archive volume. When using VM, you must respond with the virtual device number. Unless you have entered your own CMS FILEDEF command before starting the application server, the virtual device number for database archives is 181. The default virtual device number for log archives is 183. If you are taking a log archive to disk, you do not have to know the virtual device number. You can direct the archive to disk during the log archive, or change the FILEDEF command for the log archive file before starting the application server. For information on directing the log archive to disk, see your system administrator, or refer to the description of archiving procedures in the *DB2 Server for VM System Administration*. If TAPEMGR=Y, the tape must already be mounted and the tape manager handles the assign of the tape address.

When you enter `SQLEND NORMAL`, `ARCHIVE`, `LARCHIVE`, or `UARCHIVE`, the application server displays a message that indicates the number of agents or agent structures still active. An agent structure is the internal representation for a user. As agents become inactive, the application server issues another message with an updated count.

If you do a quick stop using the `SQLEND QUICK` command, the application server immediately inactivates all agents and agent structures. It then displays return code 508. You can enter an `SQLEND QUICK` command at any time, even after issuing an `SQLEND` with another keyword.

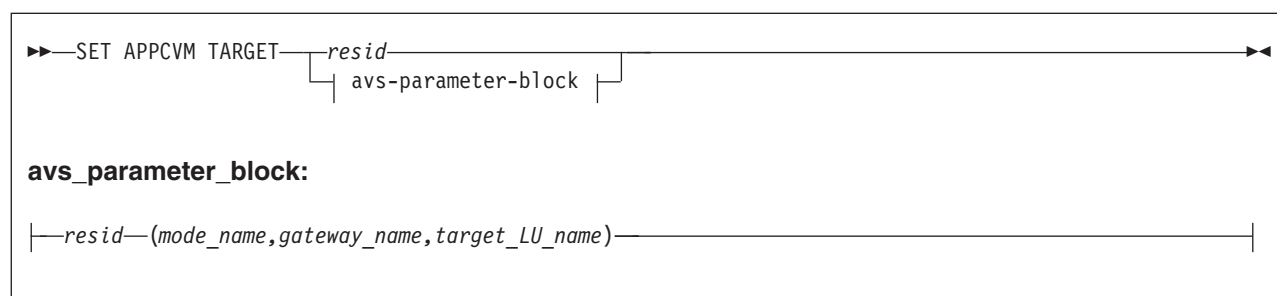
Chapter 2. Managing Online Support

This chapter explains the CIRB, CIRD, CIRT, CIRA, CIRR, and CIRC transactions, the CMS SET APPCVM command, and the DAXP and DAXT transactions used to manage DRDA2 TRUE support.

Starting VSE Guest Sharing Support

To give a VSE guest access to a VM application server, specify the SET APPCVM commands in the VSE IPL procedure. The SET APPCVM command determines the application server that the VSE guest accesses. Online and batch users can access any application server.

The following diagram illustrates the syntax of the SET APPCVM command:



In the command, the variables have the following meanings:

resid is the resource identifier of the application server. This name is the same as the *resid* variable on the IUCV *IDENT entry in the application server directory on VM.

avs_parameter_block specify these variables only when you want the VSE guest to access an application server in an SNA network. Specify the parameters in the order shown above. For more information on specifying for the AVS parameters, see the *VM/ESA: CP Command and Utility Reference*.

VSE Guest Sharing Operator Responsibilities

If you have a VSE guest sharing a VM application server, you can use all DB2 Server for VM operator commands to monitor and control VSE guest usage. You must enter these commands from the DB2 Server for VM operator console on VM. You can also resolve in-doubt logical units of work.

In-doubt logical units of work can occur if a VSE guest online user is committing a logical unit of work when the application server or CICS ends abnormally. CICS tries to automatically resolve in-doubt logical units of work. There are times, however, when you may want to manually resolve in-doubt logical units of work, such as when CICS is unable to automatically resolve the in-doubt unit of work or if you cannot wait for CICS to resolve the in-doubt unit of work.

For more information about resolving in-doubt logical units of work, see “Resolving CICS In-Doubt Logical Units of Work” on page 152.

Starting the Online Support

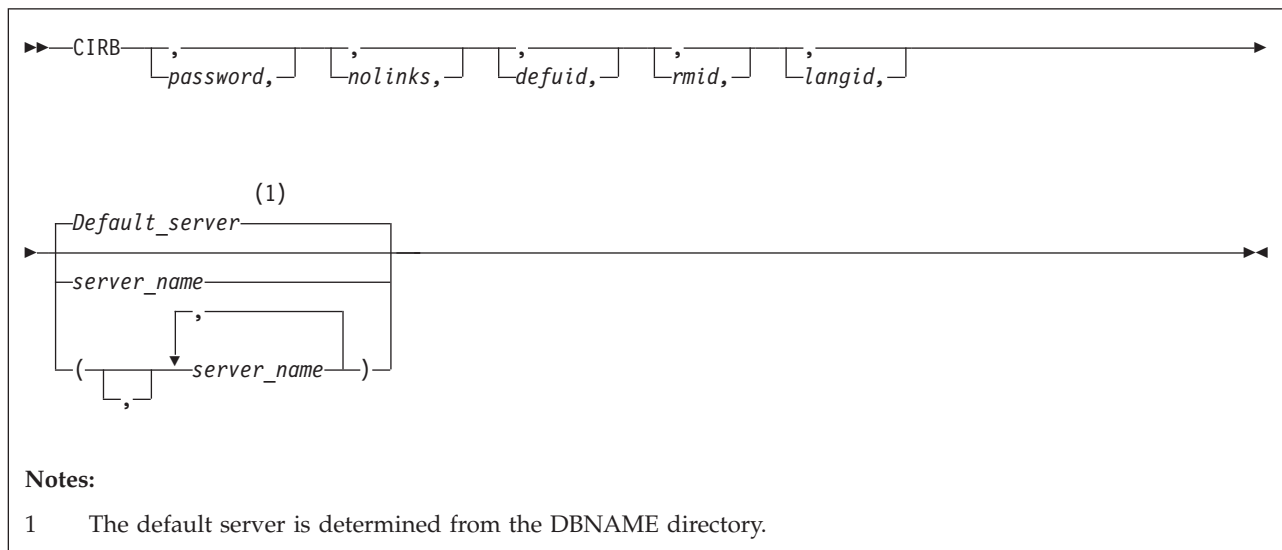
The online support allows the users at CICS terminals to communicate with the application server. A major part of the online support is the online resource adapter, which manages the communications between online (CICS) users and the application server.

The CICS sequential device support can be used to automatically initiate the CIRB transaction. See the *DB2 Server for VSE System Administration*, or *DB2 Server for VM System Administration* for more information about automatically initiating the CIRB transaction.

If the CIRB transaction has not been automatically initiated, to start the online support:

1. Ensure that the database manager is running.
2. Ensure that CICS is running.
3. Run the CICS transaction named CIRB on a CICS terminal.

CIRB has six parameters:



The parameters are described below.

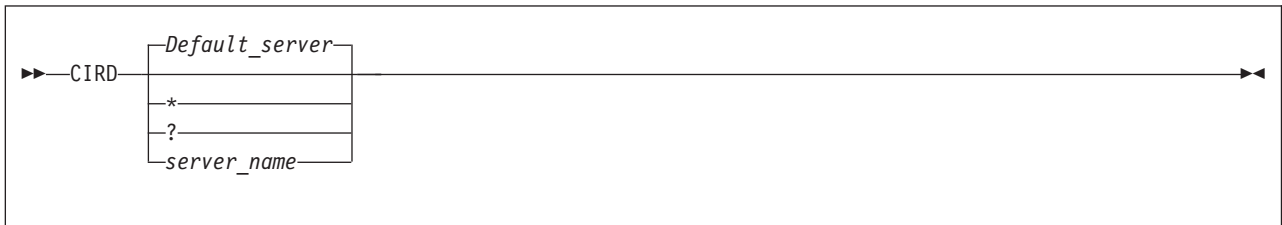
Note: If the CIRB transaction ends abnormally, enter a CIRT QUICK transaction to end the online resource adapter before re-issuing a subsequent CIRB transaction.

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	<p>This parameter establishes the operator's authority to activate online access to the application server. The password identifies the CICS subsystem. The user ID of the subsystem is the CICS APPLID, which defaults to DBDCCICS. The procedure ARIS080D uses the following job control to give the password and user ID to the application server.</p> <pre>// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S, X LOGMODE=N,PROGNAME=ARIDBS' CONNECT SQLDBA IDENTIFIED BY SQLDBAPW; GRANT SCHEDULE TO DBDCCICS IDENTIFIED BY CICSPSWD; COMMIT WORK;</pre> <p>The password chosen (CICSPSWD above) must satisfy DB2 Server for VSE specifications for a password. This password establishes which password to use when dropping connections through the CIRR or CIRT commands.</p>
NOLINKS (positional parameter 2)	3	<p>This parameter establishes the number of links (paths) that should be initialized to the local application server or remote application server connected via DRDA over TCPIP. Specify this parameter as a decimal value between 1 and 64. The number must be less than or equal to the value assigned to the NCUSERS initialization parameter of the application server. (The NCUSERS default is 5.)</p>
DEFUID (positional parameter 3)	CICSUSER	<p>This parameter identifies the default user ID used by the online support when it makes an implicit CONNECT to a local application server. This parameter must satisfy DB2 Server for VSE specifications for a user ID.</p>
RMID (positional parameter 4)	0	<p>This parameter identifies a unique resource adapter. You must specify it only if your installation has multiple CICS partitions active in the same VSE/ESA system, and if each CICS partition allows online access to the application server. For this case, recovery requires that the application server know the resource adapter it is servicing. You must specify this parameter as a decimal value between 0 and 63.</p> <p>If the online support detects that this ID is not unique in the system, it issues a message. The CIRB transaction then ends without enabling the resource adapter.</p> <p>There can be only one resource adapter enabled in a single CICS partition. An attempt to enable a second resource adapter causes the online support to issue a message, and the CIRB transaction ends without enabling the second resource adapter. The first one, however, remains in effect.</p>

Parameter	Default	Description
LANGID (positional parameter 5)	specified at installation	<p>This parameter defines the language the application server uses to display error and information messages. The language you specify on this transaction becomes the default language for ISQL, CBND, DSQG, DSQU, DSQQ and DSQD. The ISQL welcome logo always appears in the language specified on this transaction.</p> <p>This parameter must take the form of a minimum 1-character, maximum 5-character language ID (in VM, this is consistent with CMS language IDs). In the case where a national language is not supported by CMS, but it is supported by the database manager, the database manager has adopted its own langid for the language. You must use one of the language IDs in the LANGID column of the SQLDBA.SYSLANGUAGE table. The language ID must identify a language you have installed on the database manager. To choose another language, use the SET LANGUAGE command in ISQL. The following IDs can be specified on the CIRB transaction:</p> <p>AMENG American English UCENG Uppercase English FRANC French GER German KANJI Kanji (Japanese) HANZI Simplified Chinese</p> <p>If this parameter is omitted, the language defaults to the language chosen as the default at installation.</p>
SERVER_NAME (positional parameter 6)	Determined from DBNAME directory or "SQLDS."	<p>This parameter enables you to specify the application servers that you want to access. If the list format specifies multiple servers, the first one in the list becomes the default server. Only the first server_name in the list may be omitted.</p> <p>If this parameter (or the first one in the list) is omitted, the default server is determined from the DBNAME directory. If the DBNAME directory does not specify a default server, then SQLDS becomes the default server name.</p>

Displaying the Status of the Online Support

To display status information about active CICS transactions that access the application server, enter the CICS CIRB transaction. The CIRB transaction has one parameter. It operates only when the application server has been installed on CICS with the Restart Resynchronization option.



The parameter has the following meaning:

Parameter	Default	Description
SERVER_NAME (positional parameter 1)	Determined by CIRB or CIRC transaction.	This parameter enables you to specify the application server whose status is to be displayed, or * to display the status of all servers and the details of transactions accessing the servers, or ? to display a list of the connected servers without the transaction details. If this parameter is omitted, the default server_name is the one that was determined by the CIRB or the CIRC transaction.

You can enter the CIRD transaction from any CICS terminal. You can use it before a DB2 Server for VSE application is invoked to determine the active CICS transactions that access the application server. You can use it to determine the effects on users of disabling online support or shutting down the application server.

The abnormal termination process of online support automatically invokes the CIRD transaction. CIRD output is displayed on your operator console (the system operator console) after the termination error message.

The information that the CIRD transaction displays for transactions that relate to a local application server varies depending on the types of transactions in process when CIRD is invoked:

- The Dbname of the connected server in VSE/ESA only.
- Transactions waiting to access the application server.

These transactions have issued an SQL request but are waiting because all links to the application server are busy. For these transactions, CIRD displays the elapsed time of the wait.

In general, links to the application server are busy because other terminal users are accessing the application server. However, when online support is being initialized, the reason for the busy links might be that the database manager (in a VM/ESA system), or the application server (in a VSE/ESA system) and CICS logs (in either system) are being resynchronized. Resynchronization of the two logs usually requires very little time. However, the delay might be longer if resynchronization causes the rollback of a very large logical unit of work.

- Transactions currently accessing the application server.

These transactions have established a link to the application server and a logical unit of work. The application server is also currently processing that logical unit of work. For these transactions, the CIRD transaction displays the elapsed time of the current SQL statement and the elapsed time that the link has been held. The latter indicates the elapsed time of the current logical unit of work.

- Transactions holding a link to the application server but not using it.

These transactions have established a link to the application server and a logical unit of work, but the application server is not currently processing that logical unit of work. Instead, these transactions are doing other work or are waiting for a response from the CICS terminal user. For these transactions, CIRD displays the elapsed time since the last application server access was finished.

The CIRD transaction also displays the elapsed time that the link has been held. Again, this value indicates the elapsed time of the current logical unit of work.

- Transactions that previously held a link to the application server, but currently do not.

These transactions have finished one or more logical units of work but have not yet started another. For these transactions, CIRD displays the time elapsed since the last logical unit of work completed.

For CIRD to display information about a transaction, the transaction must issue an SQL request. The CIRD transaction displays the following information (where applicable) for transactions that relate to a local application server:

- The CICS task number (TASKNO).
- The CICS transaction identifier (TRANID).
- The CICS terminal identifier (TERMIN).
- The server_name of the application server connected by the online resource adapter.

Not all transactions have a terminal identifier. For example, ISQL has a two-transaction structure. One transaction (default name ISQL) controls the terminal; the other (default name CISQ) is for access to the application server. Because the CISQ transaction has no terminal associated with it, CIRD does not display TERMIN for the CISQ transactions. For CISQ, CIRD displays the terminal identifier in another field called USERDATA. USERDATA is described below.

Thus, if a transaction accesses the application server, but does not have a terminal associated with it, CIRD does not display TERMIN.

- The user identifier (USERID) that the application server establishes for the transaction.

The CIRD transaction does not display this identifier unless an authorization ID has been established. The authorization ID is established when an application issues an SQL statement that starts an initial logical unit of work. Thus, the authorization ID might not be established immediately. (For example, a transaction might be waiting for a link to the application server.) The authorization ID remains established after a transaction ends a logical unit of work, unless the RELEASE option of COMMIT WORK or ROLLBACK WORK was used.

- User data (USERDATA) for ISQL transactions.

USERDATA contains the terminal identifier (TERMIN) of the terminal that was used to invoke ISQL. For most other transactions, USERDATA is blank. You can code an online application to initialize USERDATA. Such an application would use the online cancel support.

- The elapsed time intervals (as described above).

The CIRD transaction uses the following format to display time:

hh:mm:ss

where hh is hour, mm is minutes, and ss is seconds.

After displaying the above information, CIRD displays the time of day and date. The CIRD transaction then ends its processing. The format of the displayed line is:

```
TIME=hh:mm:ss DATE=mm/dd/yy (or dd/mm/yy)
```

In DATE, mm is month, dd is day, and yy is year. The format of the date depends on the way it was specified with the DATE parameter of the VSE STDOPT job control command or statement.

If CIRD determines that no CICS transactions relate to the application server, the CIRD transaction displays only the time and date, and then ends.

Note: When the connection between an application server and the online support ends abnormally, the CIRD transaction is invoked automatically. The CIRD transaction is invoked implicitly to display information about transactions that were accessing the application server at the time of the failure. This information is displayed on the operator console or CICS terminal where the CIRD transaction was entered.

For the following examples, assume that SQLMACH1 is the default local application server and that connections have been established for the local application servers SQLMACH1, SQLMACH2 and SQLVM.

Figure 7 shows an example of the information displayed by the CIRD transaction with no parameters.

```

2 cird
F2 002 The default server is SQLMACH1.
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH1.
F2 002 Status of online DB2 Server for VSE applications:
F2 002
F2 002 Transactions waiting to establish a link to the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  WAIT TIME
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000033  MKE2           L222      JIM      L222      00:01:32
F2 002 000025  INV      L224      JIM      L222      00:08:32
F2 002
F2 002 Transactions holding a link and now accessing the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME USED      TOTAL LUW
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002           FOR CURRENT TIME
F2 002           ACCESS
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002 000019  CISQ           DEPT222  L199      L199      00:01:32      00:03:48
F2 002 000037  INV      L209      TERRY      L199      00:00:01      00:00:03
F2 002
F2 002 Transactions holding a link to the application server but not using are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE      TOTAL LUW
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002           LAST ACCESS  TIME
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002 000003  CISQ           WILLIAM  L210      L210      00:07:01      00:10:56
F2 002
F2 002 Transactions which previously accessed the application server (not holding link):
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE      TOTAL LUW
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002           LAST ACCESS  TIME
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002 000003  MKE2           ROBERT   L210      L210      00:20:04
F2 002
F2 002 TIME=14:28:23 DATE=09/01/96

```

Figure 7. Example of CIRD with Defaults

Figure 8 on page 24 shows an example of the information displayed by the CIRD transaction with a server_name specified.

```

2 cird sqlmach2
F2 002 The default server is SQLMACH1.
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH2.
F2 002 Status of online DB2 Server for VSE applications:
F2 002
F2 002 Transactions waiting to establish a link to the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  WAIT TIME
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000033  MKE2           L222     00:01:32
F2 002 000025  INV    L224    JIM           00:08:32
F2 002
F2 002 Transactions holding a link and now accessing the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME USED  TOTAL LUW
F2 002                                     FOR CURRENT  TIME
F2 002                                     ACCESS
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002 000019  CISQ           DEPT222  L199     00:01:32  00:03:48
F2 002 000037  INV    L209    TERRY           00:00:01  00:00:03
F2 002
F2 002 Transactions holding a link to the application server but not using are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE  TOTAL LUW
F2 002                                     LAST ACCESS  TIME
F2 002 -----  -----  -----  -----  -----  -----  -----
F2 002 000003  CISQ           WILLIAM  L210     00:07:01  00:10:56
F2 002
F2 002 Transactions which previously accessed the application server (not holding link):
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE
F2 002                                     LAST ACCESS
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000003  MKE2           ROBERT   L210     00:20:04
F2 002
F2 002 TIME=14:28:23 DATE=09/03/96

```

Figure 8. Example of CIRD with Server_Name

Figure 9 on page 25 shows an example of the information displayed by the CIRD transaction with the * specified.

2 cird *

```
F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 There are connections to server SQLMACH2.
F2 002 There are connections to server SQLVM.
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH1.
F2 002 Status of online DB2 Server for VSE applications:
F2 002
F2 002 Transactions waiting to establish a link to the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  WAIT TIME
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000033  MKE2             L222      00:01:32
F2 002 000025  INV      L224    JIM              00:08:32
F2 002
F2 002 Transactions holding a link and now accessing the application server are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME USED      TOTAL LUW
F2 002                                     FOR CURRENT  TIME
F2 002                                     ACCESS
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000019  CISQ             DEPT222  L199     00:01:32     00:03:48
F2 002 000137  INV      L209    BOB              00:17:34     01:24:03
F2 002
F2 002 Transactions holding a link to the application server but not using are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE     TOTAL LUW
F2 002                                     LAST ACCESS   TIME
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000013  CISQ             LARRY    L210     00:03:01     00:11:36
F2 002
F2 002 Transactions which previously accessed the application server (not holding link):
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE     LAST ACCESS
F2 002 -----  -----  -----  -----  -----  -----
F2 002 000003  MKE2             LOUISA   L210     01:57:04
F2 002
F2 002 TIME=14:28:23 DATE=09/03/96
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH2.
F2 002 There are no active application server transactions.
F2 002
F2-002 TIME= 14:29:47 DATE= 09/03/96
F2 002 -----
F2 002 DBDCCICS connected to server SQLVM.
F2 002 There are no active application server transactions.
F2 002
F2 002 TIME=14:30:23 DATE=09/03/96
```

Figure 9. Example of CIRD with *

Figure 10 on page 26 shows an example of the information displayed by the CIRD transaction with the ? specified.

```
2 cird ?
F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 There are connections to server SQLMACH2.
F2 002 There are connections to server SQLVM.
F2 002 -----
```

Figure 10. Example of CIRD with ?

Some extra information can be derived from the displays. In Figure 10 notice that SQLMACH1 is mentioned as the default server and on the next message that there are connections to SQLMACH1 also. It is possible, with the CIRR transaction, to remove the connections to SQLMACH1. The CIRD command would still show that the default server is SQLMACH1 but the message indicating there are connections to SQLMACH1 would not be displayed. In this scenario, users connecting to the default server would receive SQLCODE = -940 on the CONNECT statement. The CIRA transaction could be used to establish connections to SQLMACH1 again or the CIRC transaction could be used to change the default server to one of the other active servers. Either method allows CONNECT statements to access the default server.

If CIRR or CIRT has been issued to disconnect a server or to shut down the online resource adapter but cannot complete because there are still active transactions against the server, the CIRD transaction will show which transactions and which servers are affected.

Figure 11 on page 27 shows an example of the information displayed by the CIRD transaction with the ? parameter specified. The attempt to remove the connections to SQLMACH2 fails because there are still active transactions. Then the CIRD transaction determines which transactions are still active. The user is found and asked to complete his work. When the CIRR command is retried it completes successfully and the connections to SQLMACH2 are shut down.


```

2 cird ?
F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 There are connections to server SQLMACH2.
F2 002 There are connections to server SQLVM.
F2 002 -----
2 cirr ,,1,sqlmach2
F2 002 ARI0463I The DISABLE transaction CIRR must delay for a
          1-second interval before attempting the disable.
F2-002
2 cird ?
F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 Connections to SQLMACH2 are being disabled.
F2 002 There are connections to server SQLVM.
F2 002 -----
F2-002
2 cird *
F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 Connections to SQLMACH2 are being disabled.
F2 002 There are connections to server SQLVM.
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH1.
F2 002 There are no active application server transactions.
F2 002
F2 002 TIME= 19:07:43 DATE= 09/20/96
F2-002
F2 002 -----
F2 002 DBDCCICS connected to server SQLMACH2.
F2 002 Status of online DB2 Server for VSE applications:
F2 002
F2 002 Transactions holding a link to the application server but not using are:
F2 002
F2 002 TASKNO  TRANID  TERMID  USER ID  USERDATA  TIME SINCE  TOTAL LUW
F2 002                                     LAST ACCESS TIME
F2 002 _____  _____  _____  _____  _____  _____  _____
F2 002 0000129  CISQ      _____  CICSUSER  L77D      00:00:31  00:00:31
F2 002
F2 002 TIME= 19:07:44 DATE= 09/20/96
F2 002 -----
F2 002 DBDCCICS connected to server SQLVM.
F2 002 There are no active application server transactions.
F2 002
F2 002 TIME= 19:07:45 DATE= 09/20/96
F2-002
2 cirr ,,2,sqlmach2
F2-002 ARI0455I Connections to SQLMACH2 are disabled.

```

Figure 11. Example of CIRD in a Disable Scenario

Figure 12 on page 28 shows an example of the information displayed by the CIRD transaction.

```

F4 004 DBDCCIS connected to server SQLDB1_TOR_INV.
F4 004 Status of online application server applications:
F4 004
F4 004 Transactions waiting to establish a link to the application server:
F4 004 TASKNO TRANID TERMID USERID USERDATA WAIT TIME
F4 004 -----
F4 004 000033 MKE2                L222      00:01:32
F4 004 000025 INV      L224  JIM                00:08:32
F4 004
F4 004 Transactions holding a link and now accessing the application server:
F4 004 TASKNO TRANID TERMID USERID USERDATA TIME USED TOTAL LUW
F4 004                                FOR CURRENT TIME
F4 004                                ACCESS
F4 004 -----
F4 004 000019 CISQ                DEPT222 L199      00:01:32  00:03:48
F4 004 000037 INV      L209  TERRY                00:00:01  00:00:03
F4 004
F4 004 Transactions holding a link to the application server and not using it:
F4 004 TASKNO TRANID TERMID USERID USERDATA TIME SINCE TOTAL LUW
F4 004                                LAST ACCESS TIME
F4 004 -----
F4 004 000003 CISQ                WILLIAM L210      00:07:01  00:10:56
F4 004
F4 004 Transactions which previously accessed the application server (not holding a link):
F4 004 TASKNO TRANID TERMID USERID USERDATA TIME SINCE TOTAL LUW
F4 004                                LAST ACCESS
F4 004 -----
F4 004 000003 MKE2                ROBERT  L210      00:20:04
F4 004
F4 004 TIME=14:28:23 DATE=03/23/96

```

Figure 12. Typical Information Displayed by the CIRD Transaction

The CIRD transaction displays the following information (where applicable) for transactions that relate to a remote application server:

RDBMS

displays the name, class, and release level (version, release, and modification level) of the application server being accessed.

LU

displays the logical unit name.

TPN

displays the transaction program name. Its character and hexadecimal versions are both displayed.

TASKNO

displays the number of the task.

TRANID

displays the transaction id.

TERMID

displays the name of the terminal where the transaction was initiated.

USER ID

displays the connected user id.

STATUS

displays the communication state. COMM indicates that the transaction sent an SQL statement to the database machine and has been waiting for a reply since the time shown. APPL indicates that the transaction returned control to the application at the time shown. VRA indicates that the Online Resource Adapter is processing your request. WAIT indicates that the transaction is waiting for a session.

TIME

displays the time when the STATUS displayed had begun. For example, task number 25 has already returned control to the application at 12:17:03, as indicated by TIME.

LUWID

displays the logical unit of work identifier, which uniquely identifies an LU6.2 or TCP/IP conversation. For LU6.2 (SNA) conversation, its value is *netid.luname.instance_number.sequence_number*. For TCP/IP conversation, its value is *ip_address.port_number.instance_number.sequence_number*. The *netid* and *luname* and *ip_address* and *port_number* are up to 8 characters long, *instance_number* is 12 characters long, and *sequence_number* is 4 characters long.

Figure 13 shows an example of the information displayed by the CIRD transaction with a remote server-name specified.

```
User: 2 cird sqlmach8
System: F2 0002 The default server is SQLMACH8.
F2 0002 -----
F2 0002 Status of online DB2 Server for VSE applications for
F2 0002 RDBMS = SQLMACH8 SQLDS/VM V6.1.0
F2 0002 LU = VMC3
F2 0002 TPN = SQLMACH8
F2 0002 (X'07F6C4C2')
F2 0002
F2 0002 TASKNO  TRANID  TERMID  USER ID  STATUS  TIME
F2 0002 -----  -----  -----  -----  -----  -----
F2 0002          LUWID
F2 0002
F2 0002 0000891  DRT1   D080   SYSA     APPL    1998-08-11.09:12:42
F2 0002
F2 0002          CAIBMOML.D08001.E31FE596ADDE.0001
F2 0002
F2 0002
F2 0002 TIME= 09:18:11 DATE= 08/11/98
F2-0002
```

Figure 13. Example of CIRD with remote server name

Figure 14 shows an example of the information displayed by the CIRD transaction with a ? specified, where online access to the remote server RMTSERV1 is allowed. Assume that SQLMACH1 is the default local application server and RMTSERV1 is a remote application server. Connections have been established for SQLMACH1 and online access to RMTSERV1 through the online support is allowed.

```
User: 2 cird ?
System: F2 002 The default server is SQLMACH1.
F2 002 There are connections to server SQLMACH1.
F2 002 Online access to remote RMTSERV1 is allowed.
F2 002 -----
```

Figure 14. Example of CIRD with ?

Stopping the Online Support

To end online support, enter the CICS CIRT transaction. The CIRT transaction has three parameters:

► CIRT [password,] [mode,] [interval] ◄

After the CIRT transaction is over, online access to the application server is not possible. The CIRT parameters are positional. Syntax rules are explained in “Using Syntax Rules for DB2 Server for VSE CICS Transactions” on page 37.

The parameters have the following meanings:

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	This password establishes the operator’s authority to stop the online access to the application server. It must be the same password that was supplied for the CIRA or CIRB transaction. See “Password Implications on Online Resource Adapter Termination” on page 32 for more details.
MODE (positional parameter 2)	NORMAL	This parameter establishes the shutdown mode: NORMAL or QUICK. When remote application servers are accessed by the online support, CIRT NORMAL will complete only when all conversations to the remote application servers are deallocated. When you specify NORMAL, the CIRT transaction prevents new online users from accessing the application server. Users who are already doing work, however, can finish. When all users complete their work, no online users can use the application server. When you specify QUICK, online access to local application servers is ended immediately. Online users accessing a local application server cannot finish their work. Their current logical units of work are rolled back (unless they are already processing a COMMIT WORK). You can change from NORMAL to QUICK. However, once the MODE is QUICK, you cannot change it back to NORMAL. When remote application servers are accessed by the online support and you specify QUICK, online access to the remote application server is <i>not</i> ended immediately. Online users accessing a remote server can finish their unit of work, but cannot start a new logical unit of work.

Parameter	Default	Description
INTERVAL (positional parameter 3)	30 (seconds)	<p>The number of seconds that the CIRT transaction should delay before freeing the terminal. The value must be an integer value between 0 and 3 600. This parameter controls the availability of the CICS terminal (or operator console) after you enter the CIRT transaction.</p> <p>The CICS terminal (or VSE operator console) used to activate the CIRT transaction is unavailable until the transaction ends. This could be a long time if the online application is long-running or if a user left without correctly ending the terminal session. If you enter CIRT PASSWORD,NORMAL, the terminal is not available until all online users complete their work. Even with CIRT PASSWORD, QUICK there may be some delay before the CICS terminal allows the CIRT terminal to complete its clean-up process.</p> <p>The value you specify here represents an interval of time measured in seconds. If the CIRT transaction does not finish immediately, it waits the amount of time you specify. When this time ends, the CIRT transaction tries again to finish processing. If the CIRT transaction does not finish successfully, you receive a message telling you to retry the CIRT transaction later. After issuing the message, the CIRT transaction ends. The shutdown mode is still in effect (the application server is in the process of shutting down), and the terminal is available for your use.</p>

If links to multiple application servers exist, or if online access to multiple remote application servers exist they will all be removed. After all the links and/or online access have been removed, the online support is terminated.

The following example assumes that SQLVM, SQLMACH1 and SQLMACH2 are local application servers, and SQLMACH8 is a remote application server.

```

msg f2
AR 015 1I40I READY
2 cirt
F2-002 ARI0455I Connections to SQLVM are disabled.
F2-002 ARI0455I Connections to SQLMACH2 are disabled.
F2-002 ARI0455I Connections to SQLMACH1 are disabled.
F2-002 ARI0455I Online access to SQLMACH8 is disabled.
F2-002 ARI0413I Resource Adapter ARI00LRM is disabled.

```

Figure 15. Example of CIRT with Connections to Four Applications Servers

Note that the message “ARI0413I Resource Adapter ARI00LRM is disabled” is not displayed until the last application server connections and/or access have been severed.

When the online resource adapter is not active, the CIRA and CIRB transactions are incorrect. The online resource adapter needs to be enabled with the CIRB transaction before the CIRA and CIRB transactions can be used.

```

F2-002 ARI0413I Resource Adapter ARI00LRM is disabled.
2 cira ,,,sqlmach1
F2-002 ARI0411I Resource Adapter is not enabled.
2 cirr ,,,sqlmach1
F2-002 ARI0411I Resource Adapter is not enabled.

```

Figure 16. Example of CIRA and CIRR after CIRT

Password Implications on Online Resource Adapter Termination

The password used on the CIRR and CIRT transactions must be the same one that was used on the CIRA and CIRB transactions. CIRR and CIRT will only shut down the connections to servers where the password matches. If the passwords do not match, that server is not shut down.

Consider the following example:

1. The online resource adapter is started with the command:
CIRB pw1,5,,,,(SQLMACH1,SQLMACH2)
2. Connections to two new servers are added with the command:
CIRA ,,,(SQLMACH3,SQLMACH4)
3. Another connection is added to a fifth server with the command:
CIRA pw2,1,,SQLMACH5

It is not possible to end the online resource adapter with one command in this scenario. The CIRT or CIRR transactions must be run at least three times before the online resource adapter is completely shutdown because three different passwords were used to start it up.

The CIRT transaction entered with no parameters would only shut down the connections to SQLMACH3 and SQLMACH4 because they were the only servers that were started with the default password.

To shut down SQLMACH5, you would have to enter the following command:

```
CIRT pw2
```

To bring down the remaining servers and stop the online resource adapter you need to enter:

```
CIRT pw1 followed by CIRT
```

The CIRR transaction can also be used, but the server names must be specified. The following shows the CIRR commands that would be equivalent to the CIRT commands in this scenario.

```

CIRT pw1 is equivalent to CIRR pw1,,,(SQLMACH1,SQLMACH2)
CIRT is equivalent to CIRR ,,,(SQLMACH3,SQLMACH4)
CIRT pw2 is equivalent to CIRR pw2,,,SQLMACH5

```

If the command:

```
CIRR ,,,(SQLMACH1,SQLMACH2,SQLMACH3,SQLMACH4,SQLMACH5)
```

were entered only SQLMACH3 and SQLMACH4 would be disconnected.

Message ARI0464E will be issued for servers SQLMACH1, SQLMACH2 and SQLMACH5 because the passwords do not match.

Similarly, if the command:

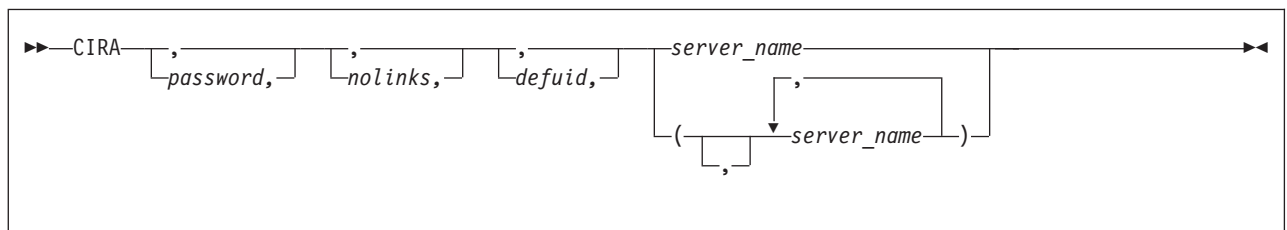
```
CIRR pw1,,,(SQLMACH1,SQLMACH2,SQLMACH3,SQLMACH4,SQLMACH5)
```

were entered only SQLMACH1 and SQLMACH2 would be disconnected.

Message ARI0464E will be issued for servers SQLMACH3, SQLMACH4 and SQLMACH5 because the passwords do not match.

Adding Online Connections Support

To add connections to another application server, issue the CICS CIRA transaction. The CIRA transaction has four parameters:



The parameters are described below. For the syntax rules of the transactions, see “Using Syntax Rules for DB2 Server for VSE CICS Transactions” on page 37.

The parameters have the following meanings:

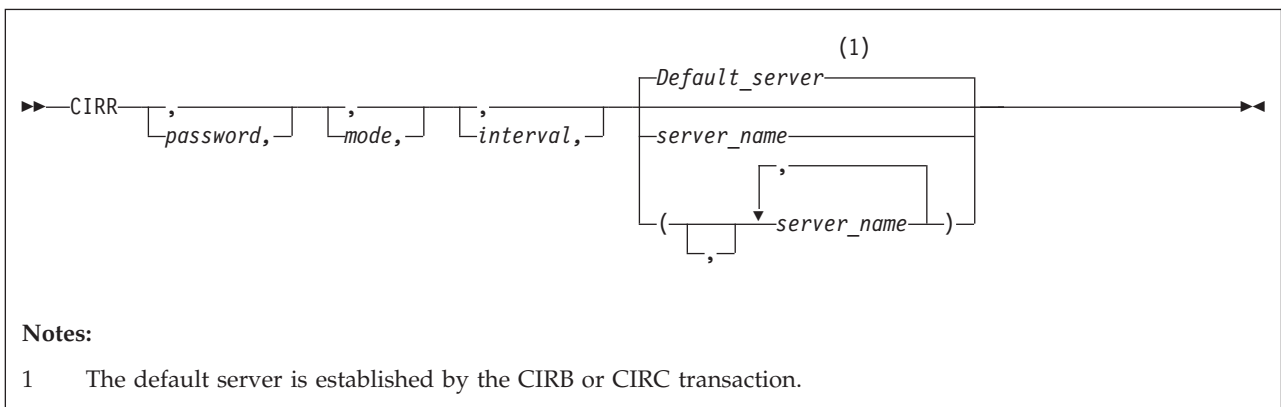
Note: The PASSWORD, NOLINKS, DEFUID and SERVER_NAME parameters have the same meanings as on the CIRB command. One exception is that the SERVER_NAME parameter is required on CIRA, but is optional on CIRB.

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	This parameter establishes the operator’s authority to activate online access to the application server. The password identifies the CICS subsystem. The user ID of the subsystem is the CICS APPLID, which defaults to DBDCCICS. The procedure ARIS080D uses the following job control to give the password and user ID to the application server. // EXEC ARISQLDS,SIZE=AUTO,PARM=' SYSMODE=S,LOGMODE=N,PROGNAME=ARIDBS' CONNECT SQLDBA IDENTIFIED BY SQLDBAPW; GRANT SCHEDULE TO DBDCCICS IDENTIFIED BY CICSPSWD; COMMIT WORK; The password chosen (CICSPSWD above) must satisfy DB2 Server for VSE specifications for a password. This password establishes which password to use when dropping connections through the CIRR or CIRT commands. See “Password Implications on Online Resource Adapter Termination” on page 32 for more details.
NOLINKS (positional parameter 2)	3	This parameter establishes the number of links (paths) that should be initialized to the local application server or remote application server connected via DRDA over TCPIP. Specify this parameter as a decimal value between 1 and 64. The number must be less than or equal to the value assigned to the NCUSERS initialization parameter of the application server. (The NCUSERS default is 5.)

Parameter	Default	Description
DEFUID (positional parameter 3)	CICSUSER	This parameter identifies the default user ID used for the implicit CONNECT of the online support to a local application server. This parameter must satisfy DB2 Server for VSE specifications for a user ID.
SERVER_NAME (positional parameter 4)	none	This parameter is required and it specifies the additional application servers that you want to access. If this parameter is omitted, the message ARI0400E is issued indicating that an incorrect input parameter was entered.

Removing Online Connections Support

To remove connections from an application server, enter the CICS CIRR transaction. The CIRR transaction has four parameters:



The parameters are described below. For the syntax rules of the transactions, see “Using Syntax Rules for DB2 Server for VSE CICS Transactions” on page 37.

The parameters have the following meanings:

Note: The PASSWORD, MODE and INTERVAL parameters have the same meanings as on the CIRT command.

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	This parameter establishes the operator’s authority to stop the online access to the application server. It must be the same password that was supplied for the server by the CIRB or CIRA transaction. See “Password Implications on Online Resource Adapter Termination” on page 32 for more details.

Parameter	Default	Description
MODE (positional parameter 2)	NORMAL	This parameter establishes the shutdown mode: NORMAL or QUICK. When you specify NORMAL, the CIRR transaction prevents new online users from accessing the specified application server. Users who are already doing work, however, can finish. When all users complete their work, no online users can use the specified application server. When you specify NORMAL for a remote application server, the shutdown of the access to the remote application server will complete only when all conversations to the remote application server have been deallocated. When you specify QUICK for a local application server, online access is ended immediately. Online users cannot finish their work. Their current logical units of work are rolled back (unless they are already processing a COMMIT WORK). You can change from NORMAL to QUICK. However, once the MODE is QUICK, you cannot change it back to NORMAL. When you specify QUICK for a remote server, the QUICK mode is changed to NORMAL.
INTERVAL (positional parameter 3)	30 (seconds)	<p>The number of seconds that the CIRR transaction should delay before freeing the terminal. The value must be an integer value between 0 and 3600. This parameter controls the availability of the CICS terminal (or operator console) after you issue the CIRR transaction.</p> <p>The CICS terminal (or VSE operator console) used to activate the CIRR transaction is unavailable until the transaction ends. This could be a long time if the online application is long-running or if a user left without correctly ending the terminal session. If you enter CIRR PASSWORD,NORMAL, the terminal is not available until all online users complete their work.</p> <p>The value you specify for <i>interval</i> represents an interval of time measured in seconds. If the CIRR transaction does not finish immediately, it waits the amount of time you specify. When this time ends, the CIRR transaction tries once again to finish processing. If the CIRR transaction does not finish successfully, you receive a message telling you to retry the CIRR transaction later. After issuing the message, the CIRR transaction ends. The shutdown mode is still in effect (the application server is in the process of shutting down), and the terminal is available for your use.</p>
SERVER_NAME (positional parameter 4)	Determined by CIRB or CIRC transaction.	This parameter enables you to specify the application servers that you want to remove access. The default server is removed if this parameter is omitted, or if the first parameter in the server_name list is blank. The default server is the one that was established by the CIRB transaction or by the CIRC transaction.

The CIRD transaction may be used to display the default server_name in case the user does not know what the default server_name is.

```

msg f2
AR 015 1I40I READY
2 cirr
F2-002 ARI0455I Connections to SQLMACH1 are disabled.
F2-002 ARI0460W Connections to the default server SQLMACH1 have
                been disabled.

```

Figure 17. Example of CIRR with Defaults

The above example assumes that there are connections and/or access to more than one server when the CIRR transaction is entered.

If the password, mode and interval are the same then the server_name list can be used to remove connections from multiple application servers. Since SQLVM was the last active connection, the online resource adapter was terminated.

```
msg f2
AR 015 1I40I READY
2 cirr ,,,(sqlmach2,sqlmach8,sqlvm)
F2-002 ARI0455I Connections to SQLMACH2 are disabled.
F2-002 ARI0455I Online access to SQLMACH8 is disabled.
F2-002 ARI0455I Connections to SQLVM are disabled.
F2-002 ARI0413I Resource Adapter ARI00LRM is disabled.
```

Figure 18. Example of CIRR with Server_Name List

The CIRR transaction can be used to remove the connections to a local application server that was established by the CIRB and CIRA transactions. It can also be used to disable online access to a remote application server. If CIRR removes the last active connections to the online resource adapter and all active APPC conversations known to the online resource adapter are deallocated, then the online resource adapter is terminated. The CIRB transaction would have to be used to restart it.

The CIRA and CIRR transactions can be entered repeatedly and in any order to add and remove links to application servers as required.

If CIRR is entered to remove connections to a server to which no connections have been established, the message "ARI0456I Connections to <server_name> do not exist." is displayed.

If the password given on the CIRR transaction does not match the password that was used to start the connections to the named server, then the connections to that server are not shut down and processing continues with the next server in the list.

Changing Online Default Support

To dynamically change the default application server, enter the CICS CIRC transaction. The CIRC transaction has one parameter:

►►—CIRC—server_name—◄◄

The parameters are described below. For the syntax rules of the transactions, see "Using Syntax Rules for DB2 Server for VSE CICS Transactions" on page 37.

The parameter has the following meaning:

Parameter	Default	Description
SERVER_NAME (positional parameter 1)	none	This parameter is required and it specifies the application server that you want to become the default. If this parameter is omitted, the message ARI0400E is issued indicating that an incorrect input parameter was entered.

The server_name specified must already have connections established to it, either from the CIRB or CIRA transactions. If connections to the specified server do not exist, message *ARI0456I Connections to <server_name> do not exist.* is displayed. In this case the CIRA transaction must first be run to establish the connections, then the CIRC transaction is run to make it the default server.

For the following example assume that connections exist to SQLMACH1 and SQLMACH2 and that SQLMACH2 is the current default server.

```
msg f2
AR 015 1I40I READY
2 circ sqlmach1
F2-002 ARI0459I The new default server is SQLMACH1.
                The previous default server was SQLMACH2.
```

Figure 19. Example of CIRC

For this next example assume that connections exist to SQLMACH1 but not to SQLMACH2.

```
msg f2
AR 015 1I40I READY
2 circ sqlmach2
F2-002 ARI0456I Connections to SQLMACH2 do not exist.
2 cira ,,sqlmach2
F2-002 ARI0454I Connections to SQLMACH2 established.
                RMCV at 0055D2E0.
2 circ sqlmach2
F2-002 ARI0459I The new default server is SQLMACH2. The previous
                default server was SQLMACH1.
```

Figure 20. Example of CIRC

It is important to note that if the connections to the default server are lost, that server is still identified as the default server. The connections can be lost because the server went down or because the CIRB transaction was used to end the connection. Users that are trying to connect to the default server in these cases will receive `SQLCODE = -940`. If the CIRB or CIRA transaction establishes connections to a server that is not ready, the message *“ARI0418A DB2 Server for VSE server <server-name> is not ready. Retry the enable transaction <tran> after DB2 Server for VSE starts.”* is displayed. If there is no active online resource adapter the CIRB transaction must be used. If there is an active online resource adapter the CIRA transaction must be used.

Using Syntax Rules for DB2 Server for VSE CICS Transactions

The syntax rules for the CIRB, CIRT, CIRA, CIRR, CIRC, CIRD, DAXP and DAXT transactions are:

- Parameters are positional:
 - For CIRB-- PASSWORD, NOLINKS, DEFUID, RMID, LANGID, SERVER_NAME
 - For CIRT-- PASSWORD, MODE, INTERVAL
 - For CIRA-- PASSWORD, NOLINKS, DEFUID, SERVER_NAME
 - For CIRR-- PASSWORD, MODE, INTERVAL, SERVER_NAME
 - For CIRC and CIRD-- SERVER_NAME
 - For DAXP-- PASSWORD, LANGID, SERVER_NAME

For DAXT-- PASSWORD, MODE, INTERVAL, SERVER_NAME

- PASSWORD and DEFUID can be up to 8 characters.
- SERVER_NAME can be up to 18 characters.
- You must separate all parameters with a blank or comma. A comma causes a specific parameter to use the default value.
- At least one blank must follow the transaction ID. (For example, CIRBSQLDBAPW is not valid.)
- DB2 Server for VSE CICS transactions default to the following:
 - If you specify only the transaction ID, all the parameters use default values.
 - If you specify only the first parameter, the rest of the parameters use default values.
 - If you specify only the first and second parameters, the rest of the parameters use default values (and so on).

You can also use the default values for parameters in other positions by placing blanks in the parameter position. Blanks indicate an unspecified parameter. The following are examples of CIRB and CIRD transactions with unspecified parameters.

In this transaction, the password defaults to SQLDBAPW:

```
CIRB ,5,SMITH,2,AMENG,SERVER_NAME
```

In this transaction, the mode defaults to NORMAL:

```
CIRT SQLDBAPW, ,30
```

In this transaction, the user ID defaults to CICSUSER:

```
CIRB SQLDBAPW,5, ,2,AMENG,SERVER-NAME
```

In this transaction, the interval defaults to 30 seconds:

```
CIRT SQLDBAPW,QUICK
```

In this transaction, the national language defaults to the language defined as the default when you installed the database manager.

```
CIRB SQLDBAPW,5,SMITH,2, ,SERVER_NAME
```

Note: No data can follow the last parameter. This transaction results in an error:

```
CIRB SQLDBAPW,2,SMITH,0,FRANC,SERVER_NAME,1234
```

Managing DRDA2 Task-Related User Exits

DB2 Server for VSE servers enable distributed unit of work support when the parameters **RMTUSERS=*n***, where $n \geq 1$, and **SYNCPNT=Y** are specified at startup. By specifying **SYNCPNT=Y**, the DB2 Server for VSE server can participate in “multiple-site read multiple-site update” logical units of work. This requires the DRDA task-related user exit to interface with the CICS/VSE Syncpoint Manager. However if **SYNCPNT=N** is specified or the DRDA task-related user exit program is not enabled, then the AXE transaction does not interface with the CICS/VSE Syncpoint Manager. This restricts the DB2 Server for VSE server to participate in “multiple-site read single-site update” units of work, where the DB2 Server for VSE server may be one of the multiple sites where data is only read, or it may be the single site where data is updated.

Once a DB2 Server for VSE application server is initialized and ready to accept connections from application requesters, then any DRDA2 DUOW application requester (such as DDCS Version 2.3, DB2 for MVS Version 3.1 and DB2 for OS/400 Version 2.3) may distribute units of work to that DB2 Server for VSE application server.

Note: If `SYNCPNT=Y` has been specified but the DRDA task-related user exit program is not enabled, the following message is issued on the DB2 Server for VSE operator console:

```
ARI0172E  A Sync Point Manager is not available.  
ARI0176I  The SYNCPNT parameter has been reset to N.
```

and DB2 Server for VSE runs as if the `SYNCPNT` startup parameter had been set to "N".

Setting Parameters Required to Autostart DRDA TRUE

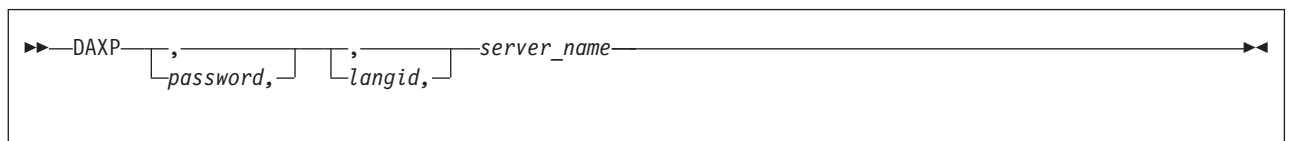
The DAXP transaction is a CICS transaction used to set the parameters which will be used when an AXE transaction subsequently autostarts DRDA TRUE support for an application server.

When the DAXP transaction is re-issued for the same application server, the new parameters will replace the old parameters. The new values will take effect the next time DRDA TRUE support is re-enabled for the application server.

The parameters set by DAXP will be effective until CICS is recycled. When CICS is recycled, it is necessary to redo DAXP for each server, if applicable. This is different from the way the CIRB or CIRA parameters are handled by the local Online Resource Adapter. In the local Online Resource Adapter, the CIRB or CIRA parameters are effective until CICS is recycled or CIRT is issued, whichever comes first.

When an AXE transaction autostarts DRDA TRUE support for an application server before the corresponding DAXP transaction was invoked, the AXE transaction will use the default `PASSWORD` and the default `LANGID`.

This transaction has three parameters.



The parameters are described below. For the syntax rules of the transactions, see "Using Syntax Rules for DB2 Server for VSE CICS Transactions" on page 37.

The parameters have the following meanings:

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	This password identifies the CICS subsystem password. It establishes the AXE transaction's authority to activate DRDA support within CICS on behalf of a DB2 for VSE application server. When the AXE transaction autostarts DRDA TRUE support, it connects to a DB2 for VSE recovery agent in order to obtain a list of LUWs that the database manager must resynchronize with CICS/VSE, it specifies a userid and password. The userid is the CICS APPLID. The password must match the password for the CICS APPLID as defined to DB2 for VSE. If not specified, the password defaults to SQLDBAPW.
LANGID (positional parameter 2)	specified at installation	This parameter defines the language the DB2 Server for VSE server uses to display error and information messages. This parameter must take the form of a minimum 1-character, maximum 5-character language ID. You must use one of the language IDs in the LANGID column of the SQLDBA.SYSLANGUAGE table. The language ID must identify a language you have installed on the DB2 Server for VSE server. The following IDs can be specified on the DAXP transaction: AMENG American English UCENG Uppercase English FRANC French GER German KANJI Kanji (Japanese) HANZI Simplified Chinese If this parameter is omitted, the language defaults to the language chosen as the default at installation.
SERVER_NAME (positional parameter 3)	none	This parameter is required and specifies the application server for which Distributed Unit of Work support is to be enabled. If this parameter is omitted, the message ARI0400E is issued indicating that an invalid input parameter was entered.

The following messages may be displayed:

1. When DAXP is re-issued for a server:

```
ARI4016I The following DRDA TRUE support enabling parameters
have been replaced:
LANGID = langid.
SERVER_NAME = server_name.
SERVER_APPLID = server_applid.
```

2. When DAXP ended successfully:

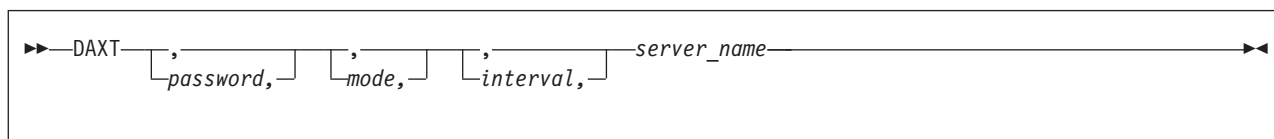
```
ARI4010I The following DRDA TRUE support enabling parameters
have been saved:
LANGID = langid.
SERVER_NAME = server_name.
SERVER_APPLID = server_applid.
These parameters will be used by the AXE transaction
when it autostarts DRDA TRUE support.
```

Disabling DRDA TRUE Support Within CICS/VSE

The DAXT transaction is a task-related user exit administration routine that is responsible for disabling DRDA TRUE support within CICS/VSE. It performs functions similar to the CIRR or CIRT transactions to disable a task-related user exit. It issues the EXEC CICS DISABLE command to disable the task-related user exit program.

This transaction is invoked by the task-related user exit program if the task-related user exit support must be disabled for some reason (e.g., a severe communication error occurs). It can also be invoked explicitly by the operator. It should be used to terminate the task-related user exit program after you have finished using it.

This transaction has four parameters.



The parameters have the following meanings:

Parameter	Default	Description
PASSWORD (positional parameter 1)	SQLDBAPW	This password establishes the operator's authority to terminate DRDA support within CICS on behalf of a DB2 for VSE application server. It must be the same password that was used to start the DRDA TRUE support. If not specified, the password defaults to SQLDBAPW.
MODE (positional parameter 2)	NORMAL	This parameter establishes the shutdown mode: NORMAL or QUICK. When you specify NORMAL, the DAXT transaction will not disable the DRDA TRUE when an AXE transaction which has accessed the DRDA TRUE earlier is still active. When you specify QUICK, the DAXT transaction will disable the DRDA TRUE immediately, unless an SQL COMMIT or ROLLBACK is ongoing.
INTERVAL (positional parameter 3)	30 (seconds)	<p>The number of seconds that the DAXT transaction should delay before freeing the terminal. The value must be an integer value between 0 and 3 600. This parameter controls the availability of the CICS terminal (or operator console) once you issue the DAXT transaction.</p> <p>The CICS terminal (or VSE operator console) used to activate the DAXT transaction is unavailable until the transaction ends. This could be a long time if the database manager is taking a long time to process the AXE transaction's Prepare-to-Commit, Commit or Rollback SQL request.</p> <p>The value you specify for <i>interval</i> represents an interval of time measured in seconds. If the DAXT transaction does not finish immediately, it waits the amount of time you specify. When this time ends, the DAXT transaction tries once again to finish processing. If the DAXT transaction does not finish successfully, you receive a message telling you to retry the DAXT transaction later. After issuing the message, the DAXT transaction ends. The shutdown mode is still in effect, and the terminal is available for your use.</p>

Parameter	Default	Description
SERVER_NAME (positional parameter 4)	none	This parameter is required and specifies the application server from which Distributed Unit of Work support is to be removed. If this parameter is omitted, the message ARI0400E is issued indicating that an invalid input parameter was entered.

The following messages may be displayed:

1. When the task-related user exit program successfully disables:
 ARI0191I *TRUE-name* is disabled.
2. When the task-related user exit program is currently not enabled, therefore, disable is not done:
 ARI0192I *TRUE-name* is not currently enabled for DRDA support.
3. When the DAXT transaction must delay before it retries to finish the DAXT process:
 ARI4012I The disable transaction *tran* must delay for a
 interval- second interval before it retries
 EXEC CICS DISABLE for DRDA TRUE *TRUE-name*.
4. When the operator must retry shutdown later:
 ARI4013A Retry the disable transaction *tran*.
 Application transactions are accessing the DRDA TRUE for
 server_name.

Chapter 3. Operating the Application Server

This chapter explains activities usually performed by the DB2 Server for VSE & VM operator:

- Issuing operator commands
- Handling VSE accounting output
- Archiving the database
- Creating a user archive
- Archiving the log
- Forcing users
- Choosing a default national language for messages
- Monitoring the application server

Issuing DB2 Server for VSE & VM Operator Commands

You can enter DB2 Server for VSE & VM operator commands in multiple user mode only. When you enter an operator command, a completion message or an error message is always given.

From the VSE System Operator Console

You can enter any DB2 Server for VSE operator command from the system console.

1. Before entering an operator command, you must enter the VSE operator MSG command. (See *VSE/ESA System Control Statements* for information on the MSG command.)
2. You are prompted to enter a DB2 Server for VSE & VM operator command. Enter the desired command and it is processed.
3. To enter another command (or reenter an incorrect command), you must again enter the VSE MSG command. If you specify a partition that is running in single user mode, the supervisor rejects the VSE Operator MSG command.
4. It is also possible to enter the desired command in the DATA= field of the MSG command. In this case, you will not be prompted to enter a DB2 Server for VSE & VM operator command. The desired command will be processed immediately. For example, MSG F4,DATA=SHOW CONNECT can be entered as a single command.

From the VM System Console

More than one application server can be running at the same time. Each application server has its own operator console, the DB2 Server for VM operator console. The user who operates this console is the DB2 Server for VM operator.

You access the operator console by logging onto the application server. The operator uses this console to perform I/O operations (such as archives and archive restores), and to monitor the operation of the application server.

When more than one application server is active, you can have more than one DB2 Server for VM operator. VM facilities, however, allow one person to operate more than one application server. For example, you can operate many application servers by running the virtual machines disconnected. You can reconnect to the various machines as required. You can also use the VM Single Console Image Facility and the programmable operator facility to run more than one application server from a single operator console.

If you want to have the VM system console serve as the DB2 Server for VM operator console, use the existing VM facilities. After initialization, disconnect the database machine and route console communications to the VM system console (or any other virtual machine) using the Single Console Image Facility (SCIF). For information on disconnecting from the database machine, see “Disconnecting the Application Server” on page 11. For additional information on the SCIF, refer to the *VM/ESA: Planning and Administration*. If you are a VM/ESA user, see the *VM/ESA CP Planning and Administration for 370* or *VM/ESA: Planning and Administration* manual.

To use SCIF, the VM directory console control statement for the database machine must identify a virtual machine as the secondary user to receive DB2 Server for VM messages. See the *VM/ESA: Planning and Administration* manual for more information about the console control statement.

The system operator can enter DB2 Server for VSE & VM commands by entering the CP SEND command. See the *VM/ESA: CP Command and Utility Reference* for information on the CP SEND command.

From the Application Server Operator Console

You can enter any operator command from the DB2 Server for VM operator console. After initialization finishes, a message is issued indicating that it is ready for operator communications. At this point, you can enter any appropriate DB2 Server for VM operator command. Whenever the application server has processed an operator command, it issues a message indicating that processing is complete, and then the next operator command can be processed.

If you stack commands, they must be valid DB2 Server for VM operator commands or responses. The application server treats any unexpected response (because of an incorrectly ordered stack) as an incorrect command or message reply. The application server continues reading and processing stack entries until there are no more entries in the stack. To stack commands, type the LINEND character, usually #, between each command on the same command line. For example, to stack the SHOW SYSTEM, SHOW POOL, and SHOW USERS commands, enter the following:

```
show system#show pool#show users
```

You can use the CP command Q TERMINAL to determine your setting of the LINEND character.

From a CICS Terminal

The DB2 Server for VSE product provides an optional component called Interactive Structured Query Language (ISQL) that allows CICS terminal users to access an application server. ISQL uses CICS for terminal control and runs as a CICS transaction.

A user can enter the SHOW and COUNTER operator commands in ISQL. You use these commands to monitor system performance; this chapter describes them.

The ISQL SHOW and COUNTER commands can be used when the target application server is either a local or a remote DB2 Server for VSE & VM

application server. For information on using ISQL, consult the *DB2 Server for VSE & VM Interactive SQL Guide and Reference* manual.

From User Displays

The DB2 Server for VM product provides a component called Interactive Structured Query Language (ISQL) that allows CMS terminal display users to access an application server. The ISQL facility runs as an application of CMS.

Terminal display users can enter the SHOW and COUNTER operator commands through the ISQL facility; you use these commands to monitor system performance. While you can enter the SHOW and COUNTER commands from the operator console when you are using DRDA protocol, you cannot use them with the ISQL facility. The *DB2 Server for VSE & VM Interactive SQL Guide and Reference* manual explains the ISQL facility.

DB2 Server for VSE Accounting

Your procedures for operating the application server might include the use of DB2 Server for VSE accounting. You enable accounting by specifying ACCOUNT=T,E, or D upon initialization. The default, ACCOUNT=N, indicates that accounting is not done. (ACCOUNT is an initialization parameter.)

If ACCOUNT=T, the application server writes accounting records to a tape file identified in the job control for startup. If ACCOUNT=D, the application server writes accounting records to a DASD file, which you must identify in the job control. If ACCOUNT=E, the application server writes accounting records to a VSAM ESDS file, which you must identify in the job control.

If you start the application server and direct accounting output to a tape file, you are requested to mount a tape and you are prompted for the cuu of the tape drive. (When accounting information is directed to a DASD file, no prompt is necessary.)

When accounting to tape, the application server avoids overlaying the previous session's accounting information by not rewinding the accounting tape at OPEN or CLOSE time. The next time you startup, another file is written starting after the previous session's file. When you direct accounting information to DASD, the problem of overlaying a previous session's file exists only if stand-alone sequential access method (SAM) manages the file. If stand-alone SAM manages the DASD file, you should refer to the *DB2 Server for VSE System Administration* manual for information about setting up job control and operating procedures that avoid the problem.

When the application server ends either normally or abnormally, it attempts to close the accounting file. If the accounting file is on tape and, for some reason, the accounting file cannot be closed, you should manually write a tape mark. Use the VSE MTC WTM command to write a tape mark. The *VSE/ESA System Control Statements* manual describes the MTC WTM command.

Using Alternate Accounting Files

Your procedures for accounting might also include the use of DB2 Server for VSE alternate accounting file support. The alternate accounting file support lets you switch to an alternate accounting file in two ways: by using the DB2 Server for VSE ALTACCT operator command, or by requesting the use of an alternate file in response to an error message.

The job control for starting the application server must include job control for two accounting files for the application server to use the alternate accounting file support. Both files must be either on tape or on DASD. (You cannot have one file defined on tape and the other on DASD.) The file name on the TLBL or DLBL statement for the first accounting file is ARIACC1. The file name on the TLBL or DLBL statement for the second accounting file is ARIACC2. The job control for the accounting files should already be in the job control procedure that identifies the application server. If the job control has not been included, see "Setting Up Job Control for the Accounting Files" in the *DB2 Server for VSE System Administration* manual.

You must switch to the alternate accounting file when:

- You direct the accounting output to DASD, and you receive an end-of-extents message.
- A write error occurs on the current accounting file for which you receive an error message.
- You want to close the current file (perhaps for processing) but do not want to shut down the application server.

In the first two cases, you do not need to enter the ALTACCT command. When the application server detects the error, it issues a message asking whether you want to:

1. Continue processing with the accounting disabled
2. Switch to the alternate accounting file and continue running
3. End.

Aside from the error conditions, the other reason you would want to switch accounting files is to close the current file without stopping the application server. Then one accounting file can be processed while operation continues. In this situation, enter the ALTACCT command. Note that ALTACCT has no parameters.

```
▶▶—ALTACCT—◀◀
```

When you start the application server to use accounting, the ARIACC1 file is always opened first. When you request that the application server use the alternate file, it closes the ARIACC1 file and opens the alternate file (ARIACC2).

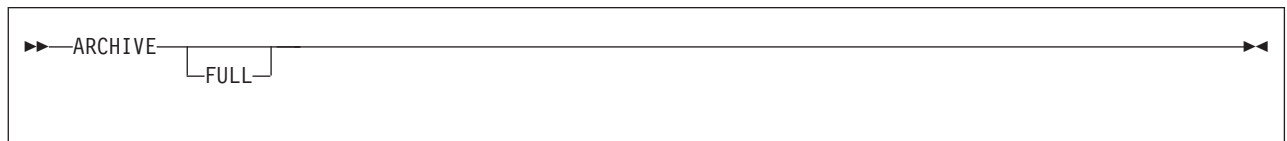
If you direct the accounting output to tape, the application server prompts you for the cuu of the alternate file. The cuu of the first accounting file is unassigned. Thus, until you restart, the cuu of the first accounting file is available for any purpose, including reuse for an alternate accounting file.

If you enter ALTACCT again, the application server closes ARIACC2 and opens ARIACC1. You can switch between accounting files any number of times during a session.

Archiving the Database

There are three ways to archive a database:

- You can start the application server so that it will archive the database automatically. Specify LOGMODE=A in the DB2 Server for VSE & VM initialization parameters. The ARCHPCT parameter controls how often a database archive is created. If you do not specify ARCHPCT, the application server will use the default (80).
- You can archive the database when stopping the application server. Enter an SQLEND ARCHIVE or SQLEND UARCHIVE operator command. (See “Stopping the Application Server” on page 11 for more information on the SQLEND command.)
- You can explicitly request a database archive while the application server is running in multiple user mode. Enter the ARCHIVE operator command:



The application server must be running with either LOGMODE=A or L. Automatic archiving of either the database or the log must already be in effect before you can explicitly request a database archive. DB2 Server for VSE supports both dynamic and static tape drive assignment during the archiving process. Dynamic assign, the default, allows access to one tape drive while archiving, whereas static assign allows access to multiple tape drives. The use of static tape drive assignment can decrease the time required to archive the log or database. Startup JCL must include assign statements to permit static assign. Figure 21 shows an example of startup JCL required for using static assign.

```
// JOB SQLUSQL  DB2/VSE FROM (SQLUSR1,ECHO=N0) 02/10/96 15:44:09
// LIBDEF *,SEARCH=(PRD2.SQL340)
// TLBL ARILARC
// TLBL ARILALT
// TLBL ARIARCH
// ASSGN SYS005,cu1
// ASSGN SYS005,cu2,ALT
// EXEC ARISQLDS,SIZE=AUTO,
//          PARM='ACCOUNT=T,ARCHPCT=80,DISPBIAS=9,DSPLYDEV=B,LOGMODE*
//          =L,NCUSERS=05,STARTUP=W,SYSMODE=M'
```

Note: For cu1 and cu2, specify your installation's tape devices.

Figure 21. Example of Startup JCL for Static Tape Drive Assign Archive

When the FULL option is used on the ARCHIVE command, these archives may be used as a starting point for taking incremental archives with Data Restore. Database manager archives taken with the FULL parameter, will be the associated full backup for subsequent incremental backups taken with Data Restore.

Implicit archives occur only in LOGMODE=A or LOGMODE=L. In LOGMODE=L, the database manager will archive the active log. If alternate logging is enabled, both the inactive log and the active log are archived if the inactive log was not archived previously. In LOGMODE=A, the database manager will take a database archive.

When LOGMODE=A, the database is already being archived automatically, so it is unusual to have to enter an ARCHIVE command. One reason to enter an

ARCHIVE command would be to free log space prior to the next automatic archive. Note that when LOGMODE=A, the ARCHIVE command causes only a database archive. Log archiving is not involved.

When LOGMODE=L and alternate logging is not enabled, log archives are automatically being done, but not database archives. You may want to explicitly archive the database during long periods of operation with LOGMODE=L. The time required to restore the database increases as the frequency of log archives increases. If you enter the ARCHIVE command when LOGMODE=L, the application server does a log archive if there is information in the log, and then a database archive. If alternate logging is enabled, both the inactive log and the active log are archived if the inactive log was not archived previously. The command is rejected if a database or log archive is already in progress.

As with any archive, the application server requests that you mount the required tape volume or volumes to contain the database archive (or log archives, if LOGMODE=L, and you are not archiving the logs to a VM disk). For dynamic assign, the operator enters the virtual device number (cuu) of the tape drive when prompted by message ARI0299A; for static assign, the operator enters '1' to indicate that the archive will be written to one or more statically assigned tape drives (SYS005). If logical device SYS005 is not available, the operator will be notified by display message ARI2036I that the device is not available. The system will then re-display message ARI0299A prompting the operator to specify a valid cuu. For more information on static and dynamic tape drive assign, and the *DB2 Server for VSE System Administration* manual. The archive or archives are then created. In a VM system, when the application server prompts you to mount and ready the archive volume, you should respond with the virtual device number. Unless you have entered your own CMS FILEDEF command before startup, the virtual device number for database archives is 181. The virtual device number for log archives is 183.

Note: When LOGMODE=N, in single user mode, a log for recovery is not maintained. When LOGMODE=Y, the log is maintained but is not automatically archived.

Creating a User Archive

In VM/ESA, user archives are database archives created with non-DB2 Server for VM facilities such as the VM/ESA DASD Dump Restore Service Program (DDR). In VSE/ESA, user archives are database archives done with non-DB2 Server for VSE facilities, such as the VSE/VSAM IDCAMS Backup/Restore Feature. User archives include the DB2 Server for VSE & VM directory and all dbextents, but not the log.

Because the DB2 Server for VSE & VM code that creates the archives is DASD independent, it does not take advantage of particular DASD characteristics to improve performance. Some non-DB2 Server for VSE & VM facilities exploit these characteristics and can archive and restore your database faster in some cases.

Note: The DB2 Server for VM directory and dbextent minidisks are non-CMS minidisks.

If you are going to do a non-DB2 Server for VSE & VM archive of your database, you can only do it when you shut down. Use the following command:

```
SQLEND UARCHIVE
```

After all logical units of work finish, the application server indicates in the log that a user archive will be done. It then reminds the operator to do the archive, and ends. Figure 22 shows an example of the control statements required to archive a DB2 Server for VSE database using the VSE/VSAM IDCAMS BACKUP command.

```
// JOB USERARCH
// ASSGN SYS005,181
// DLBL IJSYSUC,'SQL301C',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  BACKUP (SQL301.BDISK, -
          SQL301.DDSK1, -
          SQL301.DDSK2, -
          SQL301.DDSK3, -
          SQL301.DDSK4, -
          SQL301.DDSK5, -
          SQL301.DDSK6, -
          SQL301.DDSK7)
/*
/ &
```

Figure 22. Example of VSE/VSAM Backup Command for a User Archive

The control statements cause VSE/VSAM to archive a database having a directory (BDISK) and seven dbextents (DDSK1-DDSK7). You do not archive the log data set(s). For more information about the VSE/VSAM BACKUP command, see the *IBM VSE/VSAM Using Commands and Macros* manual. If LOGMODE=L, the system (VSE or VM) does a log archive before ending if there is information in the log. After the application server ends, you should archive the DB2 Server for VSE & VM directory and all the dbextents, but not the logs.

If you specify STARTUP=R or U the next time you start the application server, it assumes that you have not done the user archive; otherwise, the system sends a message to the operator console asking if the user archive was done successfully.

If you decide to do user archives, you can do intermediate log archives or intermediate database archives to free log space while the application server is running.

For intermediate log archives, set LOGMODE to L when starting the application server. When LOGMODE=L and alternate logging is not enabled, the system automatically does a log archive when the log fills to the ARCHPCT value. If alternate logging is enabled, an attempt will be made to switch to the inactive log rather than forcing an archive of the active log. If you prefer to schedule your online log archives, you can periodically enter the LARCHIVE command. If alternate logging is enabled, you can use LARCHIVE INACTIVE to archive the inactive log.

For intermediate database archives, set LOGMODE to A. When LOGMODE=A, the application server automatically archives the database when the log fills to the ARCHPCT value. Note that database archiving, not user archiving, is performed. Use the ARCHIVE command if you want to schedule your online database archives. Refer to “Archiving the Database” on page 46 for more information on the ARCHIVE command.

You can use the SHOW LOG operator command to monitor available log space and to assist you in scheduling user archives. See “SHOW LOG” on page 113 for more information about the SHOW LOG command.

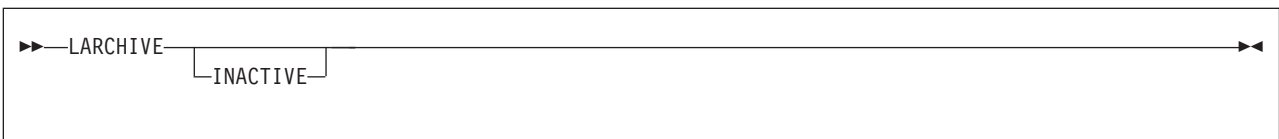
Archiving the Log

Your procedures for recovery can also include archiving the log. A log archive is a copy of the DB2 Server for VSE & VM log on either a magnetic tape or in VM only, on a disk.

There are three ways to do a log archive:

- Periodically, as part of the procedure for starting the application server.
Specify LOGMODE=L in the initialization parameters when you start the application server. If alternate logging is not enabled, the ARCHPCT parameter controls how often an archive is done. If alternate logging is enabled, an attempt to switch to the inactive log will be made when ARCHPCT is reached. If you do not specify ARCHPCT, the default value is used.
- Immediately, as part of the procedure for stopping the application server.
Enter the SQLEND LARCHIVE operator command. See “Stopping the Application Server” on page 11 for information on the SQLEND command.
- By explicit request, when the application server is running in multiple user mode.
Use the LARCHIVE operator command.

The syntax of the LARCHIVE operator command is as follows:



The application server must be running with LOGMODE=L. Automatic log archiving must be in effect before you can request a log archive explicitly. When LOGMODE=L and alternate logging is not enabled, the application server automatically does periodic log archives; therefore, it is unusual to have to enter the LARCHIVE command. One reason to enter the LARCHIVE command is to free log space before the next automatic log archive. The command is rejected if a database or log archive is already in progress.

The LARCHIVE command causes only a log archive. The application server does not archive the database as a result of the LARCHIVE command.

As with any archive, the application server requests that you mount the required tape volume to contain the log archive. The log is then copied to tape.

You can specify LARCHIVE INACTIVE only when alternate logging is enabled. This command will attempt to archive the inactive log. If it was previously archived, an error message will be displayed.

Archiving the Log to Tape in VM/ESA

When you are using a tape for the log archive, and TAPEMGR=N is specified, the application server requests that you mount the required tape volume to contain it. The log archive is then copied to tape. When the application server prompts you to

mount and ready the archive volume, respond with the virtual device number (cuu). Unless you entered your own CMS FILEDEF command before starting the application server, the virtual device number is 183.

If TAPEMGR=Y is specified, the database manager will automatically perform the tape assign.

Archiving the Log to Disk in VM/ESA

You can direct a log archive to disk by changing the file definition for the log archive file with the CMS FILEDEF command.

If you receive message ARI0246D, you can direct a log archive to disk by replying CHANGE to the message.

Forcing Users to End Their Work

To force users to end their work, you can use either the SHOW ACTIVE, SHOW SYSTEM or SHOW CONNECT ACTIVE command followed by the FORCE operator command. The SHOW CONNECT command can be used with its other parameters, except for the INACTIVE, and in VM only, the WAITING parameters, which do not enable the FORCE command. You must enter the SHOW ACTIVE, SHOW CONNECT and FORCE commands from either the DB2 Server for VM operator console or the VSE system operator console for the force to succeed.

Note: For one example of why you may be required to force users, see “Resolving CICS In-Doubt Logical Units of Work” on page 152.

The operator of an unlike application server can sever the telecommunication link at the application server site, causing a rollback. The operator of an application server can also force unlike application requesters, thus severing the communication link and causing a rollback.

Suppose you want to force user JESSICA to end her DB2 Server for VSE & VM work. First, you must determine JESSICA’s agent number. An agent is the application server’s internal representation for a user. The application server numbers each agent.

To determine JESSICA’s agent number, enter a SHOW ACTIVE command as illustrated in Figure 23 on page 51 and Figure 24 on page 52.

```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show active
F4 004 Status of agents:
F4 004   Checkpoint agent is not active.
F4 004   User Agent:  1 User ID: JONES   is R/O APPL  7BA
F4 004     Agent is   processing and is in I/O      WAIT.
F4 004   User Agent:  3 User ID: JESSICA is R/W APPL  5A4
F4 004     Agent is   not processing and is in communication WAIT.
F4 004   User Agent:  4 User ID: TESTUSER is NIW SUBS
F4 004     Agent is   not processing and is in communication WAIT.
F4 004   1      agent(s) not connected to an APPL or SUBSYS.
```

Figure 23. A Typical Example of the SHOW ACTIVE Command for Determining an Agent Number in VSE

```

show active
Status of agents:
Checkpoint agent is not active.
User Agent:  1 User ID:  SMITH   is R/W APPL  7B4
Agent is    processing and is in I/O      Wait.
User Agent:  2 User ID:  MICHAEL is R/O SUBS  7B9
Agent is    not processing and is in communication Wait.
User Agent:  3 User ID:  JESSICA IS R/O APPL  5A4
Agent is    processing an SQL/DS operator command.
User Agent:  4 User ID:  TESTUSER IS R/W APPL  7BB
Agent is    processing and is in I/O      Wait.
User Agent:  5 User ID:  MACNEILL IS R/O APPL  7B9
Agent is    not processing and is in communication Wait.
5 agent(s) not connected to an APPL or SUBSYS.
ARI0065I Operator command processing is complete.

```

Figure 24. A Typical SHOW ACTIVE Display on a VM Operator Console or ISQL Terminal

JESSICA is agent 3. At this time, you might want to note JESSICA's logical unit of work identifier. The logical unit of work identifier is the rightmost number on the line. JESSICA's is 5A4. You can use the number later to verify that the application server forced JESSICA's work.

To force JESSICA to end her work, you enter:

```
FORCE 3
```

or

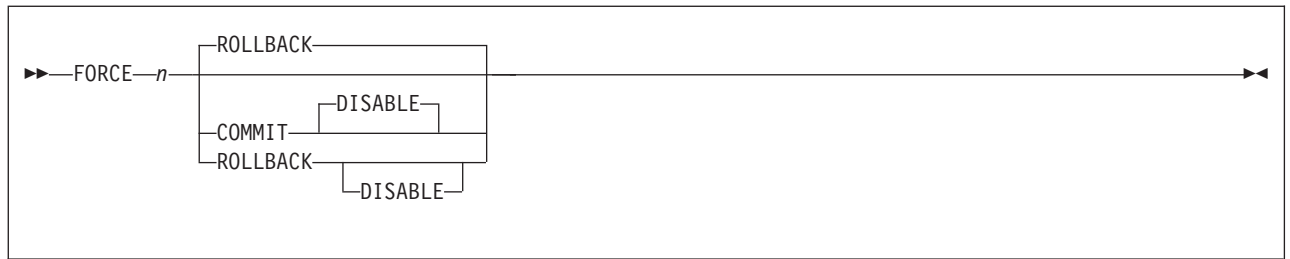
```
FORCE 3 ROLLBACK
```

To verify that the application server has forced JESSICA's work, reenter the SHOW ACTIVE command. The display either shows that JESSICA is no longer assigned to the agent or that the agent is not active. It could also show that JESSICA is still assigned to the agent, but not in work (NIW). If JESSICA started another logical unit of work, the display shows a new logical unit of work identifier.

If the system rejects the FORCE command, a SHOW ACTIVE displays results similar to those described above. JESSICA may have disconnected before the application server processed the FORCE command, or may have ended her logical unit of work.

The FORCE command does not end the application. For this purpose, use other system facilities. For example, you can use the CP FORCE command from the VM system console. If the application is from a VSE guest, you can use the VSE system operator CANCEL command. For a CICS transaction, you can use CEMT TRMNAT.

The FORCE operator command has the following syntax:



The FORCE command causes the logical unit of work identified by agent number *n* to be backed out (when you specify ROLLBACK) or committed (when you specify COMMIT). The COMMIT option is valid only for an in-doubt logical unit of work. An in-doubt logical unit of work can only come from VSE guests. (An in-doubt logical unit of work is identified by PREPARED FOR COMMIT OR ROLLBACK in the SHOW ACTIVE display.) Always specify COMMIT for ISQL in-doubt logical units of work.

If you specify the DISABLE option, the connection to the application requester is disabled, the logical unit of work is rolled back, and the application server makes the connection available for another user (after the rollback completes). You can use this option to make the agent structure available to other users.

The DISABLE option is valid for VM users and VSE guests or agents connected to batch or VSE/ICCF applications (identified by APPL in the SHOW ACTIVE display) or via TCP/IP.

You must use CICS commands to stop long-running CICS transactions. (CICS transactions are identified by SUBS in the SHOW ACTIVE display.) To break all online connections, use the CICS CIRT transaction.

When you enter the FORCE command for an in-doubt logical unit of work, the application server makes the agent structure and connection available for other users. For in-doubt units, DISABLE is implicit, and you do not have to specify it.

When you enter the FORCE command for an in-doubt agent involved in DUOW, a warning message is displayed prompting the operator to confirm the force.

When you enter a FORCE command, the application receives a negative SQLCODE (error code) for the current or next SQL request. If the application server is running in VM with SQLDS protocol when you enter the FORCE command, the application will receive either SQLCODE -933 or -948. If the application server is serving only local requesters in a VSE system, the application will receive either SQLCODE -916 or -948 when you enter the FORCE command. The code is dependent on whether a DISABLE was issued along with the FORCE. When DRDA is installed, remote users will receive SQLCODE -30080. For an explanation of these negative SQLCODES see the *DB2 Server for VM Messages and Codes* manual.

Forcing DB2 Server for VSE Remote Users to End Their Work

The FORCE RMTUSER operator command can be used to force remote (DRDA) DB2 Server for VSE users. The command can force users whether they have real agents or not. Also, the real agent may not be in a logical unit of work when the FORCE RMTUSER is initiated.

The FORCE RMTUSER operator command can also be used to free up the control structure for a remote connection. This will then allow other remote users to access the server.

Note: The FORCE RMTUSER command will roll back the current LUW and will cause the conversation to end on the next request.

The FORCE RMTUSER operator command has the following syntax:

```
▶—FORCE RMTUSER—rmtuser ID—◀
```

Before entering the FORCE RMTUSER command, a SHOW CONNECT must be entered to verify the remote user identifier (*rmtuser ID*). The output display of the SHOW CONNECT command will identify each remote user by a unique *rmtuser ID*, for example:

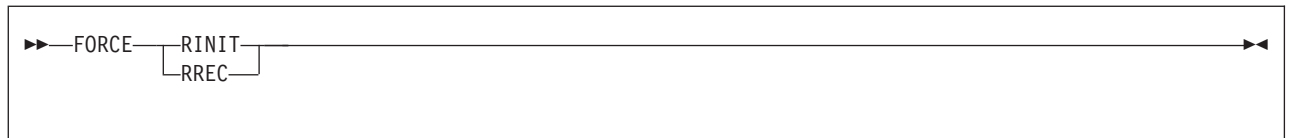
```
⋮
F4 004      User Agent:   5   User-ID: SQLUDAY1 SQL-ID: SQLUDAY1
F4 004          is R/O  APPL 14729
F4 004      Agent is processing and is in communication wait.
F4 004          State started: 1996-10-21 14:23:08
F4 004          Conversation started: 1996-10-21 14:22:57
F4 004          CPU time: 00:00:01
F4 004          LUWID: CAIBMOML.OECGW001.A679B3B27B1E.0001
F4 004          EXTNAM: SQLUDAY1.1
F4 004          Requester: SQLDS/VM V3.3.0 at TOIVMLB6
F4 004          Rmtuser ID: 2
F4 004          LU name: OMPGW001
F4 004          Task no.: 0000046
F4 004          Protocol: DRDA
⋮
⋮
```

Use of this information will help prevent an operator from mistakenly forcing the wrong remote user from the system.

Note: If the operator FORCE command does not stop the remote user successfully, the conversation can be ended by using the CICS master operator transaction (CEMT) which ends the APPC-to-XPCC Exchange Transaction (AXE) of the remote user. Forced ending of an AXE transaction by using the CEMT Terminate Task will result in a de-allocate abend to the application requester and an XPCC abnormal disconnect to the application server. For information on the use of the CEMT command, refer to the *CICS/VSE CICS-Supplied Transactions* manual.

VM/ESA FORCE Enhancements for CRR

It could occur that during *resynchronization initialization* or *resynchronization recovery*, the database manager waits indefinitely for a response from the CRR recovery server. If this is the case, the following command may be used to terminate *resynchronization initialization* processing:



If “FORCE RINIT” is entered, *resynchronization initialization* processing is terminated and the SYNCNT parameter is changed from Y to N. If “FORCE RREC” is entered, *resynchronization recovery* is terminated and deallocate (abend) is performed to terminate the conversation with the CRR recovery server.

Notes:

1. The operator must issue the SHOW ACTIVE, SHOW CONNECT or SHOW SYSTEM command prior to the FORCE RINIT/RREC command. Otherwise, the following message is issued:

ARI0225E System operator must issue SHOW ACTIVE, SHOW CONNECT or SHOW SYSTEM command prior to FORCE command.

and FORCE processing terminates.

2. If the database is not performing *resynchronization initialization* when the FORCE RINIT command is entered, then the following message is displayed:

ARI2040E FORCE RINIT may only be entered when Resynchronization Initialization is active.

and FORCE processing terminates.

3. If the database is not performing *resynchronization recovery* when the FORCE RREC command is entered, then the following message is displayed:

ARI2040E FORCE RREC may only be entered when Resynchronization Recovery is active.

and FORCE processing terminates.

4. If the FORCE RINIT command was already issued, the following message is displayed:

ARI2041E FORCE RINIT is already scheduled.

and FORCE processing terminates. (Note that the scheduled FORCE command remains!)

5. If the FORCE RREC command was already issued, the following message is displayed:

ARI2041E FORCE RREC is already scheduled.

and FORCE processing terminates. (Note that the scheduled FORCE command remains!)

6. If extra parameters are entered after “FORCE RINIT/RREC”, then the following message is displayed:

ARI0229E Too many FORCE command input parameters

and FORCE processing terminates.

Forcing DB2 Server for VM Inactive Users

The FORCE INACTIVE operator command can be used to force inactive DB2 Server for VM users. The FORCE INACTIVE operator command can force users only when they do not have real agents. The FORCE operator command must be used for active users with real agents. Active users are those users who are

connected to the database and are in a logical unit of work. They are using a real agent and can be identified by the SHOW CONNECT ACTIVE operator command. Inactive users are those users who are connected to the database, but are not in a logical unit of work. They are not using a real agent and can be identified by the SHOW CONNECT INACTIVE operator command.

The FORCE INACTIVE operator command can be used to make connections available to other users or to free connections when necessary. For example, the SQLEND command will wait until all connections are released before shutting down the database manager. If there are inactive users that are delaying this, the operator can use the FORCE INACTIVE command to release those connections and allow the SQLEND processing to continue.

The FORCE INACTIVE operator command has the following syntax:

```
►—FORCE INACTIVE—connection ID—◄
```

Before entering the FORCE INACTIVE command, a SHOW CONNECT or SHOW CONNECT INACTIVE command must be entered to verify the connection user identifier (*connection ID*). The output display of the SHOW CONNECT command will identify each user by a unique *connection ID*, for example:

```
SHOW CONNECT
Status of Connected Users                2002-02-27 15:56:05
Checkpoint agent is not active.
User ID: DB2ADMIN SQL ID: DB2ADMIN
User is inactive.
State started: 2002-02-27 15:52:48
Conversation started: 2002-02-27 15:52:48
LUWID: G91A4EBB.C107.030F27205304.0001
EXTNAM: db2bp.exe 0180018F
Requester: DB2 Connect 7.1 (DB2/NT) at
Protocol: DRDA via TCP/IP
Connection ID: 2
1 Users are active.
0 Users are waiting.
1 Users are inactive.
5 Agents are available.
43 User connections are available.
ARI0065I Operator command processing is complete.

FORCE INACTIVE 2
ARI0065I Operator command processing is complete.
```

Use of this information will help prevent an operator from mistakenly forcing the wrong inactive user from the system.

Choosing a National Language

Using the following command, you can choose a national language for messages on the operator console:

▶▶ SET LANGUAGE

language
langid

You must specify the language or a language ID. The language you choose must be one installed on the application server. If you are using VSE/ESA, you cannot choose a double-byte character set (DBCS) language, because the VSE operator console does not support DBCS.

Note: ISQL users can choose a double-byte character set (DBCS) when issuing the SET LANGUAGE command. KANJI is one example of a language that uses a double-byte character set.

Keyword	Description
language	is any language installed on the application server. You must use one of the names in the LANGUAGE column of the SQLDBA.SYSLANGUAGE catalog table. For more information on the SYSLANGUAGE table, see Appendix A of the <i>DB2 Server for VSE & VM Database Administration</i> manual.
langid	is the language ID of any language installed on the application server. You must use one of the language IDs in the LANGID column of the SQLDBA.SYSLANGUAGE table. You can also use language IDs that CMS supports. Valid language IDs are: AMENG American English UCENG Uppercase English FRANC French GER German KANJI Kanji (Japanese) HANZI Simplified Chinese

Monitoring Performance

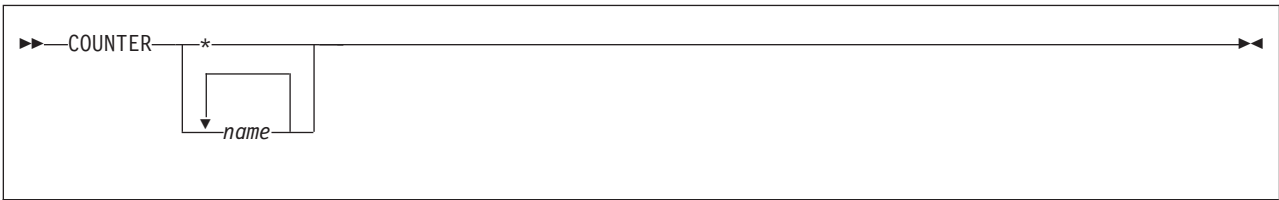
The COUNTER and SHOW commands let you monitor DB2 Server for VSE & VM performance. When VSE guests in a VM system are accessing the application server, and the application server is running with CICS, you can also get more performance data by using the CICS/VSE Monitoring Facility. See the *CICS/VSE Installation and Operations Guide* for a description of this facility.

You can use performance data to tune the application server. This manual does not give guidelines for making decisions on tuning performance; that information is in the *DB2 Server for VSE & VM Performance Tuning Handbook*.

You can also use the SHOW commands to find system bottlenecks. The *DB2 Server for VSE & VM Diagnosis Guide and Reference*, describes how to use the SHOW commands for problem determination.

You can enter the COUNTER and SHOW commands from the ISQL facility. The following sections list and describe the COUNTER and SHOW commands; each command is followed by an example of the information it displays.

COUNTER



The application server counts the occurrences of key events. To display all counters, enter:

```
COUNTER *
```

To display specific counters, list them after the command name like this:

```
COUNTER LOGREAD LOGWRITE
```

To set the counters to zero, enter the RESET command. Refer to “RESET” on page 61 for a description of the RESET command. Each counter is set to zero when you start the application server. If a counter reaches its limit (2,147,483,647) and overflows, it is reset to zero and starts over.

Below are the names of the counters and what they count.

RDSCALL is the number of Relational Data System (RDS) calls. The RDS system is a component of the application server that runs in either the VSE database partition or the VM database machine. The RDS system supports the relational model of data. The relational model permits users to see data as though it existed in tables containing rows and columns. The language that you use to access data is SQL (Structured Query Language). The RDS system translates all SQL statements into specific tasks for the Database Storage Subsystem. (See DBSSCALL.)

RDSCALL represents the number of times that application programs (including user programs, the ISQL facility, the DBS Utility, and the preprocessors) access the application server.

DBSSCALL is the number of Database Storage Subsystem (DBSS) calls. The DBSS system is a component of the application server that runs in either the VSE database partition or the VM database machine. The DBSS system is the storage manager of the application server. It provides locking mechanisms to prevent concurrent users from interfering with one another. It also provides all recovery management, database I/O, and sorting.

The DBSSCALL counter represents the number of times that the RDS system accesses the DBSS system to satisfy RDS calls.

BEGINLUW is the number of logical units of work that were started.

ROLLBACK is the number of logical units of work that were rolled back.

CHKPOINT is the number of DB2 Server for VSE & VM checkpoints taken.

LOCKLMT is the number of times the lock escalation process failed (for the number of successful escalations, see ESCALATE).

Escalation occurs when the application server tries to replace small locks acquired by a logical unit of work with a single larger lock.

(The larger lock is always a DBSPACE lock.) When an escalation fails, the usual reason is that the attempt to acquire a DBSPACE lock causes a deadlock. A logical unit of work that reaches a lock limit must be rolled back. The process could also fail when an insufficient number of locks are freed by the acquisition of the larger lock. (This rarely happens.)

LOCKLMT plus ESCALATE is the number of times that logical units of work reach the user or system lock limit. You set lock limits with the NLRBU initialization parameter. You define the lock limit for the system with the NLRBS initialization parameter.

ESCALATE is the number of times that the lock escalation process ends successfully. Escalation succeeds whenever a dbspace lock is successfully acquired to free lower level locks.

ESCALATE plus LOCKLMT is the number of times that a logical unit of work reached the user (NLRBU) or system (NLRBS) lock limit.

WAITLOCK is the number of lock requests that resulted in waits. When a user accesses data, the application server acquires a lock on that data to prevent interference from other users. If someone already has a lock on the data, the user may have to wait for the lock to be freed. Every time a user must wait to get a lock, the application server increases the WAITLOCK counter.

DEADLCK is the number of deadlocks detected. A deadlock occurs when two users are each waiting for a resource that the other has locked. Neither user can continue processing. When the application server detects a deadlock, it rolls back one of the logical units of work. The other user can then continue processing.

LPAGBUFF is the number of times the application server looked in the page buffer (looked at a dbspace header, data, or index page). The application server reads from secondary storage only if the page it requires is not in the page buffer.

PAGEREAD is the number of DBSPACE page reads.

PAGWRITE is the number of DBSPACE page writes.

LDIRBUFF is the number of times the application server looked in the directory buffer. The application server reads secondary storage only if the block required is not in the directory buffer.

DIRREAD is the number of directory block reads.

DIRWRITE is the number of directory block writes.

LOGREAD is the number of log page reads.

LOGWRITE is the number of log page writes.

DASDREAD is the total number of DASD reads (PAGEREAD + DIRREAD + LOGREAD).

DASDWRTIT is the total number of DASD writes (PAGWRITE + DIRWRITE + LOGWRITE).

DASDIO is the total DASD I/O (DASDREAD + DASDWRTIT).

LTIMEOUT is the number of lock timeouts detected. A lock timeout occurs when the application server detects that an agent has been in a

lock wait state longer than the time specified by the LTIMEOUT initialization parameter and rolls back that agents logical unit of work.

The following examples show the counters displayed when you enter **COUNTER ***.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 counter *
F4 004 Counter values at  DATE='01-10-96'  TIME='20:59:59'
F4 004 Calls to RDS                RDSCALL : 431
F4 004 Calls to DBSS              DBSSCALL: 2044
F4 004 LUWS started                BEGINLUW: 82
F4 004 LUWS rolled back           ROLLBACK: 9
F4 004 System checkpoints taken   CHKPOINT: 2
F4 004 Maximum locks exceeded     LOCKLMT : 0
F4 004 Lock escalations           ESCALATE: 0
F4 004 Waits for lock              WAITLOCK: 4
F4 004 Deadlocks detected         DEADLCK : 0
F4 004 Looks in page buffer       LPAGBUFF: 4503
F4 004 DBSPACE page reads         PAGEREAD: 370
F4 004 DBSPACE page writes       PAGWRITE: 108
F4 004 Looks in directory buffer  LDIRBUFF: 770
F4 004 Directory block reads      DIRREAD : 23
F4 004 Directory block writes    DIRWRITE: 0
F4 004 Log page reads             LOGREAD : 20
F4 004 Log page writes            LOGWRITE: 19
F4 004 Total DASD reads           DASDREAD: 413
F4 004 Total DASD writes         DASDWRT: 127
F4 004 Total DASD I/O            DASDIO  : 540
F4 004 Lock timeouts detected     LTIMEOUT: 0
F4 004 ARI0065I Operator command processing is complete.

```

Figure 25. A Typical COUNTER Display on a VSE Operator Console

```

counter *
Counter values at  DATE='01-10-96'  TIME='20:59:59'
Calls to RDS                RDSCALL : 431
Calls to DBSS              DBSSCALL: 2044
LUWS started                BEGINLUW: 82
LUWS rolled back           ROLLBACK: 9
System checkpoints taken   CHKPOINT: 2
Maximum locks exceeded     LOCKLMT : 0
Lock escalations           ESCALATE: 0
Waits for lock              WAITLOCK: 4
Deadlocks detected         DEADLCK : 0
Looks in page buffer       LPAGBUFF: 4503
DBSPACE page reads         PAGEREAD: 370
DBSPACE page writes       PAGWRITE: 108
Looks in directory buffer  LDIRBUFF: 770
Directory block reads      DIRREAD : 23
Directory block writes    DIRWRITE: 0
Log page reads             LOGREAD : 20
Log page writes            LOGWRITE: 19
Total DASD reads           DASDREAD: 413
Total DASD writes         DASDWRT: 127
Total DASD I/O            DASDIO  : 540
Lock timeouts detected     LTIMEOUT: 0
ARI0065I Operator command processing is complete.

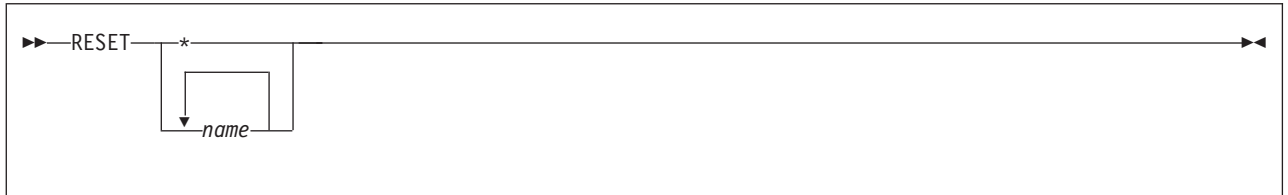
```

Figure 26. A Typical COUNTER Display on a VM Operator Console or ISQL Terminal

Note: There are several important things to remember about these counters if you have installed the optional VMDSS code to support VM Data Spaces:

- The database manager does not increment PAGEREAD or PAGWRITE when the operating system retrieves a page from DASD into a data space, writes a page from a data space to DASD, or reads or writes a page to or from the local buffers.
- It does not increment DIRREAD or DIRWRITE when the Directory is using Data Spaces Support.

RESET



The RESET command sets the named counters to zero. Specify RESET * to set all counters to zero. (See "COUNTER" on page 58 for the counter names.)

Note: You cannot reset the DASDIO, DASDREAD, and DASDWRIT counters because they are sums of other counters.

You can enter the RESET command only from the DB2 Server for VM or VSE system operator consoles.

RESET CRR LOGNAMES (VM only)

▶▶—RESET—CRR—LOGNAMES—◀◀

The RESET CRR LOGNAMES command lets the database manager reset its own log name and the CRR recovery server's TPN and log name to hex zeros. This command can be used when log name mismatch situations arise. When issued, it displays the following message:

```
ARI0173I The CRR recovery server's TPN and log name and the database
manager's log name have been reset. On the next
resynchronization, the database manager's log status
will be cold.
```

Notes:

1. This command can be executed when SYNCPT=N. This is useful to help fix problems that may arise during resynchronization initialization that cause SYNCPT to be reset to N.
2. This command cannot be executed during resynchronization initialization. If the command is issued, the following message is displayed:

```
ARI0198E Reset CRR Lognames cannot be issued when
Resynchronization Initialization is active.
```

3. This command cannot be executed during resynchronization recovery. If the command is issued, the following message is displayed:

```
ARI0198E Reset CRR Lognames cannot be issued when
Resynchronization Recovery is active.
```

RESET HIGHSTOR

▶▶—RESET HIGHSTOR—◀◀

The RESET HIGHSTOR command resets the previous high storage values to the current allocated values; it will also reset the USERID field to show the current high storage holder. The RESET HIGHSTOR command has no parameters.

The RESET HIGHSTOR command, when used in conjunction with the SHOW STORAGE command, can aid in determining the maximum storage required to perform a specific function.

The RESET HIGHSTOR command can be entered only from the VSE system operator console or the DB2 Server for VM operator console.

newval

is the new value for the specified initialization parameter.

CHKINTVL

A decimal value from 1 to 99999999. The default is 10.

DISPBIAS

A decimal value from 1 to 10. The default is 7.

DSPSTATS

A value indicating how much, if any checkpoint performance data is displayed. The default is 00, meaning no display at shutdown. The first character can have a value of 0, 1, or 2. The second character can have a value of 0 or 1. The first character indicates if checkpoint performance information is displayed. 0 indicates no display, 1 indicates checkpoint times are displayed, 2 indicates detailed checkpoint performance information is displayed. The second character indicates if counter information is displayed when the application server is shutdown. 0 indicates no display and 1 indicates counter information is displayed at shutdown.

DUMPTYPE

P, F, or N. P indicates partial virtual machine dump, F indicates full virtual machine dump, and N indicates no dump. F is the default.

LTIMEOUT

A decimal value from 0 to 99999. A value of 0 means that no LTIMEOUT is in effect. The default is 0.

PTIMEOUT

A decimal value from 0 to 99999. A value of 0 means that no PTIMEOUT is in effect. The default is 180.

PROCXAB

An integer between 0 and 255. The default is 0.

TAPEMGR

Y or N. Y indicates that a tape manager is being used and N indicates that no tape manager is being used. N is the default.

ARCHTAPE (Valid for VSE only)

UNL or REW. UNL indicates that tapes which are part of a multivolume archive will be unloaded at end of tape. REW indicates that the tapes must be manually unloaded. The default is REW. This parameter only applies to VSE. In VM, unloading at end of tape is controlled by the FILEDEFs.

TCPMAXRT

An integer value from 1 to 9999. The default is 158. For a complete description of TCP/IP support, refer to *"DB2 Server for VM System Administration, Chapter 16, Using TCP/IP with DB2 Server for VM"* or to *"DB2 Server for VSE System Administration, Chapter 15, Using TCP/IP with DB2 Server for VSE"*.

TCPRETRY

Y or N. Y indicates the database manager will try to automatically restart TCP/IP support if it fails. N indicates the database manager will not try to automatically restart TCP/IP support if it fails. Y is the default. For a complete description of TCP/IP support, refer to *"DB2 Server for VM System Administration, Chapter 16, Using TCP/IP with DB2 Server for VM"* or to *"DB2 Server for VSE System Administration, Chapter 15, Using TCP/IP with DB2 Server for VSE"*.

TCPCAN

Y or N. Y indicates that the Cancel TCP/IP Agent in VM support is enabled. N

|
|

|
|
|

indicates that the cancel support is disabled. There is some performance overhead if cancel support is enabled. This overhead may be unacceptable for some users.

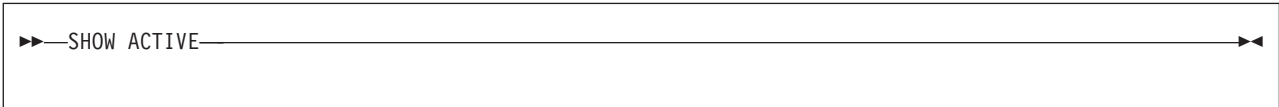
See Appendix A, “DB2 Server for VSE & VM Initialization Parameters,” on page 179 for more information on the initialization parameters.

Figure 27 is an example of the SET command.

```
ARI0045I Ready for operator communications.  
  
SET CHKINTVL 200  
ARI0065I Operator command processing is complete.  
  
SET DISPBIAS 7  
ARI0065I Operator command processing is complete.  
  
SET DSPSTATS 21  
ARI0065I Operator command processing is complete.  
  
SET DUMPTYPE F  
ARI0065I Operator command processing is complete.  
  
SET LTIMEOUT 300  
ARI0065I Operator command processing is complete.  
  
SET PTIMEOUT 180  
ARI0065I Operator command processing is complete.  
  
SET PROCMXAB 1  
ARI0065I Operator command processing is complete.  
  
SET TCPMAXRT 50  
ARI0065I Operator command processing is complete.  
  
SET TCPRETRY N  
ARI0065I Operator command processing is complete.
```

Figure 27. SET command examples

SHOW ACTIVE



The SHOW ACTIVE command displays the status of the active agent structures in the database partition or machine. An agent structure, or agent, is the internal representation of an active user. The application server uses these structures to service SQL requests by users, to process operator commands, and to perform checkpoints and database archives.

The status of certain agent structures are not displayed by the SHOW ACTIVE command. They are the operator agent, ready or recovery agent, and the TCP/IP agent if TCP/IP is active. The operator agent structure and the ready or recovery agent structure always exist in the application server. The operation agent is used for operator command processing, and it exists in both multiple and single user modes. The ready or recovery agent is only used in the VM/ESA operating system for VSE guests. It exists in multiple user mode only. The TCP/IP agent handles the coordination of TCP/IP clients. It only exists in multiple user mode and when TCP/IP is active.

Another agent structure is used only for checkpoint and database archive work. This structure is called the CHECKPOINT AGENT in the command display. The checkpoint agent always exists in both single and multiple user modes. The display shows whether the agent is dormant, waiting to start, or processing. If the checkpoint agent is waiting to start or is processing, the display also shows if it is an ARCHIVE checkpoint.

The status of each active agent structure is displayed following this information as described below:

- The agent number (agent identifier).
- The authorization ID of the person requesting the services represented by the agent structure. The authorization ID is typically the user's virtual machine user ID.

For remote connections, the DB2 Server for VM authorization ID can also be the ID specified by the *userid* tag in the communications directory file, or a non-VM user ID. If the application server is in the process of verifying the authorization ID, it is not shown. The string *null id* is displayed instead.

- The work status of the agent structure.

The SHOW ACTIVE command displays:

- NIW (not in work) when no logical unit of work is in progress
- R/O for read-only logical units of work
- R/W for read-write logical units of work

Note: If the logical unit of work is updating only data that resides in a nonrecoverable storage pool, then R/O is displayed (not R/W).

- NEW is for logical units of work that have started but have not yet accessed the application server.

NEW is applicable only when log archiving is starting. NEW units of work may not continue until the checkpoint that starts a log archive ends. See Figure 33 on page 69 for an example of NEW agent structures.

- The state of processing of the agent structure, that is, whether the agent is dormant, waiting to start, or processing. See Figure 30 on page 68 or Figure 39 on page 76 for an example of state processing information.
- When a logical unit of work is in progress, a unique internal logical unit of work identifier is displayed. (See "Forcing Users to End Their Work" on page 51 for more information on the internal logical unit of work ID.) In Figure 30 on page 68, user agents 1 and 3 are processing logical units 7B4 and 5A4, respectively.
- All active user-agent structures from VM display APPL (for DB2 Server for VM application program). User agent structures from VSE guests display APPL for batch applications, and SUBS for the CICS subsystem.
- Whether the structure is doing batch or CICS online work. For batch, APPL (batch application) is displayed; for CICS online, SUBS (representing subsystem) is displayed. In the VSE console example below, user JONES is an online user whose requests are coming from the CICS subsystem.
- Whether a rollback or commit of a logical unit of work is scheduled and the reason. SHOW ACTIVE displays ROLLBACK|COMMIT SCHEDULED if the activity has not yet started; it displays ROLLBACK|COMMIT ACTIVE if the activity is in progress.
- Whether an SQL request is currently being processed for the user. If it is, you see AGENT IS PROCESSING. If not, you see AGENT IS NOT PROCESSING.

- The reason the agent structure is waiting. The type of wait can be one of the following: communication, lock, latch, checkpoint, page buffer (OUT OF PAGE), block buffer (OUT OF BLOCK), or database I/O.

Note: An agent is in communication wait when it is waiting for an SQL request from the user, or is waiting for an APPC/VM function to end. An agent is in latch wait when it is waiting for a latch on a page or block buffer. If the agent is not waiting, it will display the number of times the database manager looks in the page buffer (LPAGBUFF) since the beginning of the logical unit of work.

- If the agent is not waiting, it will display the number of times the database manager looks in the page buffer (LPAGBUFF) since the beginning of the logical unit of work is displayed.
- The following information applies only for VSE guests:
If the application server or the CICS system had previously abnormally ended, some agents may have a display like the following:

```

User Agent      7 User ID: JONES  is prepared
for commit or rollback
VM ID= VSE1  COORDINATOR= DBDCCICS  RESOURCE ADAPTER= 0
TRANSACTION= TPSP  CICS SIGNON= PETER  TERMINAL= L012

```

Figure 28. A Typical Example of an In-doubt Logical Unit of Work in VM

```

F4 004      User Agent      7 User ID: JONES  is prepared
F4 004      for commit or rollback
F4 004      COORDINATOR= DBDCCICS  RESOURCE ADAPTER= 0
F4 004      TRANSACTION= TPSP  CICS SIGNON= PETER  TERMINAL= L012

```

Figure 29. A Typical Example of an In-doubt Logical Unit of Work in VSE

In particular, note that the user ID is followed by the message IS PREPARED FOR COMMIT OR ROLLBACK. This identifies the agent’s logical unit of work as being in-doubt. (In-doubt logical units of work occur when a system failure happens while an online user is committing a logical unit of work.) If running under the CICS subsystem, in-doubt logical units of work will be resolved automatically the next time you run the CIRB transaction. However, you may want to resolve in-doubt logical units of work manually. If you enter SHOW ACTIVE and see that some logical units of work are in-doubt, refer to “Resolving CICS In-Doubt Logical Units of Work” on page 152.

- When you enter SHOW ACTIVE or SHOW SYSTEM from ISQL, your agent number is displayed as processing an operator command. See Figure 81 on page 134 for information on the SHOW SYSTEM command.

Following the display of the status of each active agent structure, SHOW ACTIVE displays the number of inactive (NOT CONNECTED...) agent structures.

The system operator or an ISQL user can enter the SHOW ACTIVE command. An ISQL user can only enter the SHOW ACTIVE command when connected to an application server using SQLDS protocol. When you enter SHOW ACTIVE on the operator console, any agent that starts a logical unit of work becomes forceable. The agent remains forceable until the logical unit of work ends. You can use the FORCE command (described on page 51) only on forceable agents and you can

enter the FORCE command only from the operator console. The application server resets the agent's forceable indicator when the agent's logical unit of work commits or rolls back.

In VM, SHOW ACTIVE shows only the users who currently have an agent structure. To display all users who are connected to the application server and their status, the DB2 Server for VM operator or ISQL terminal user can enter the SHOW USERS command or the SHOW CONNECT command. For further information on these commands, see Figure 83 on page 137 and "SHOW CONNECT" on page 73.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show active
F4 004 Status of agents:
F4 004   Checkpoint agent is not active.
F4 004   User Agent:  1 User ID: SMITH   is R/W APPL  7B4
F4 004     Agent is   processing and is in I/O          Wait.
F4 004   User Agent:  2 User ID: MICHAEL is R/O SUBS   7B9
F4 004     Agent is not processing and is in communication Wait.
F4 004   User Agent:  3 User ID: JESSICA IS R/O APPL  5A4
F4 004     Agent is processing an SQL/DS operator command.
F4 004   User Agent:  4 User ID: TESTUSER IS R/W APPL  7BB
F4 004     Agent is   processing and is in I/O.
F4 004   User Agent:  5 User ID: MACNIELL IS R/O APPL  7B9
F4 004     Agent is not processing and is in communication Wait.
F4 004   5 agent(s) not connected to an APPL or SUBSYS.
F4 004 ARI0065I Operator command processing is complete.

```

Figure 30. A Typical SHOW ACTIVE Display on a VSE Operator Console

Figure 30 is a typical SHOW ACTIVE display. Figure 31 shows the display you might see when R/O or R/W logical units of work are delaying the checkpoint that starts a log archive. If checkpoints are delayed, the database manager delays new logical units of work until the checkpoint finishes.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show active
F4 004 Status of agents:
F4 004   Checkpoint agent is waiting to start log archive
F4 004     New logical units of work and the log archive
F4 004     process cannot continue until all R/O, R/W,
F4 004     and prepared logical units of work are completed.
F4 004   User Agent:  1 User ID: CHAN    is R/W APPL  106
F4 004     Agent is not processing and is in communication Wait.
F4 004   User Agent:  2 User ID: EWING   is NEW APPL  107
F4 004     Agent is waiting for log archive checkpoint.
F4 004   3 agent(s) not connected to an APPL or SUBSYS.

```

Figure 31. A Typical Example of Logical Units of Work Delaying a Checkpoint on VSE

```

show active
Status of agents:
Checkpoint agent is not active.
User Agent: 1 User ID: SMITH is R/W APPL 7B4
Agent is processing and is in I/O Wait.
User Agent: 2 User ID: MICHAEL is R/O SUBS 7B9
Agent is not processing and is in communication Wait.
User Agent: 3 User ID: JESSICA IS R/O APPL 5A4
Agent is processing an SQL/DS operator command.
User Agent: 4 User ID: TESTUSER IS R/W APPL 7BB
Agent is processing and is in I/O.
User Agent: 5 User ID: MACNIELL IS R/O APPL 7B9
Agent is not processing and is in communication Wait.
0 agent(s) not connected to an APPL or SUBSYS.
ARI0065I Operator command processing is complete.

```

Figure 32. A Typical SHOW ACTIVE Display on a VM Operator Console or ISQL Terminal

Figure 32 is a typical SHOW ACTIVE display. Figure 33 shows the display you might see when R/O or R/W logical units of work are delaying the checkpoint that starts a log archive. If checkpoints are delayed, the application server also delays new logical units of work until the checkpoint finishes.

```

show active
Status of agents:
Checkpoint agent is waiting to start log archive
New logical units of work and the log archive
process cannot continue until all R/O, R/W,
and prepared logical units of work are completed.
User Agent: 1 User ID: CHAN is R/W APPL 106
Agent is not processing and is in communication Wait.
User Agent: 2 User ID: EWING is NEW APPL 107
Agent is waiting for log archive checkpoint.
3 agent(s) not connected to an APPL or SUBSYS.
ARI0065I Operator command processing is complete.

```

Figure 33. A Typical Example of Logical Units of Work Delaying a Checkpoint on VM

SHOW ADDRESS

```

▶▶—SHOW ADDRESS—module-name—◀◀

```

The SHOW ADDRESS command displays the load address, compile date and PTF identifier (if it exists) of CONV, DSC, DBSS, DRRM, RDS, and WUM modules. This command can help with problem determination. You can use it for debugging. This command can also be used to determine if a PTF has been applied.

The output of this command is the hexadecimal load address of the module you specify, or an error message if the module is not found. This load address applies only to the application server, not to an application requester from which the command may have been entered (in SQLDS protocol).

Figure 34 on page 70 shows sample output from the SHOW ADDRESS command.

```
show address ariym00
ARIYM00 97.090 PLX      : 6CDBA0 HEX
```

Figure 34. A Typical SHOW ADDRESS Display on a VM Operator Console or ISQL Terminal

```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004          Enter a DB2 Server for VSE operator command.
4 show address ariym00
F4 004 ARIYM00 97.090 PLX      : 6CDBA0 HEX
```

Figure 35. A Typical SHOW ADDRESS Display on a VSE Operator Console

SHOW BUFFERS

```
▶▶—SHOW BUFFERS—◀◀
```

The database manager maintains a pool of 4096-byte buffers that contain the most frequently accessed database pages. It also maintains a set of 512-byte buffers that contain the most frequently accessed directory blocks. The SHOW BUFFERS command displays information about the buffer pool size and usage. This information can be used to tune the database manager.

Figure 36 on page 71 shows the output of the SHOW BUFFERS command.

```

AR
MSG F4
AR 015 1I40I READY
F4 004 ARI0062A SQLDS :
F4-004          Enter a DB2 Server for VSE operator command.
4 show buffers
F4 004 Only used buffers are displayed.
F4 004 DBSPACE  REC      ADDR      FLAGS  FIX CNT
F4 004 Page Buffers
F4 004 1      000090  00608000  00     0
F4 004 1      000093  00609000  00     0
F4 004 1      00014D  0060A000  00     0
F4 004 1      00014C  0060B000  80     0
F4 004 2      000094  0060C000  00     0
F4 004 2      000091  0060D000  00     0
F4 004 1      00008D  0060E000  00     0
F4 004 1      000088  0060F000  80     0
F4 004 5      00008B  00610000  00     0
F4 004 1      00008F  00611000  00     0
F4 004 2      000080  00612000  00     0
F4 004 1      000092  00613000  80     0
F4 004 1      00015A  00614000  80     0
F4 004 1      00014E  00615000  80     0
F4 004 1      00008E  00616000  00     0
F4 004 1      00008C  00617000  00     0
F4 004 32001  0000FD  00618000  00     1
F4 004 1      000156  00619000  80     0
F4 004 32001  0000F9  0061A000  00     0
F4 004 1      000155  0061B000  80     0
F4 004 1      00014A  0061C000  80     0
F4 004 1      000089  0061D000  00     0
F4 004 1      00008A  0061E000  00     0
F4 004 1      000149  0061F000  80     0
F4 004 1      00014F  00620000  80     0
F4 004 1      00014B  00621000  80     0
F4 004 Directory Buffers
F4 004 1      000001  00622000  80     0
F4 004 0      000079  00622200  80     0
F4 004 1      000002  00622400  80     0
F4 004 2      000001  00622600  80     0
F4 004 2      000002  00622800  00     0
F4 004 2      000003  00622A00  00     0
F4 004 2      000004  00622C00  00     0
F4 004 89     000001  00622E00  00     0
F4 004 0      000004  00623000  00     0
F4 004 88     000001  00623200  00     0
F4 004 ARI0065I Operator command processing is complete.

```

Figure 36. A Typical SHOW BUFFERS Display on a VSE Operator Console

```

show buffers
Only used buffers are displayed.
DBSPACE  REC      ADDR      FLAGS  FIX CNT
Page Buffers
  1      000090   00608000   00     0
  1      000093   00609000   00     0
  1      00014D   0060A000   00     0
  1      00014C   0060B000   80     0
  2      000094   0060C000   00     0
  2      000091   0060D000   00     0
  1      00008D   0060E000   00     0
  1      000088   0060F000   80     0
  5      00008B   00610000   00     0
  1      00008F   00611000   00     0
  2      000080   00612000   00     0
  1      000092   00613000   80     0
  1      00015A   00614000   80     0
  1      00014E   00615000   80     0
  1      00008E   00616000   00     0
  1      00008C   00617000   00     0
32001   0000FD   00618000   00     1
  1      000156   00619000   80     0
32001   0000F9   0061A000   00     0
  1      000155   0061B000   80     0
  1      00014A   0061C000   80     0
  1      000089   0061D000   00     0
  1      00008A   0061E000   00     0
  1      000149   0061F000   80     0
  1      00014F   00620000   80     0
  1      00014B   00621000   80     0
Directory Buffers
  1      000001   00622000   80     0
  0      000079   00622200   80     0
  1      000002   00622400   80     0
  2      000001   00622600   80     0
  2      000002   00622800   00     0
  2      000003   00622A00   00     0
  2      000004   00622C00   00     0
  89     000001   00622E00   00     0
  0      000004   00623000   00     0
  88     000001   00623200   00     0
ARI0065I Operator command processing is complete.

```

Figure 37. A Typical SHOW BUFFERS Display on a VM Operator Console or ISQL Terminal

The columns of the output of SHOW BUFFERS have these meanings:

DBSPACE is the dbspace number. If the dbspace number is 32001, the page buffer contains a log page. If the dbspace number is 0, the directory block contains a system block not related to any dbspace.

REC is the logical page number in the page buffer. For directory buffers, it is the relative page map table block number. These blocks contain logical (dbspace page number) to physical (storage pool slot) mappings for pages in the dbspace. Block 1 contains the mappings for pages 1 to 128 of the dbspace, block 2 contains the mappings for pages 129 to 256, and so on.

ADDR is the virtual address of the buffer.

FLAGS indicate the page status:

'80' Modified Page

The buffer page has been modified.

'40' In Transit The buffer page is being copied to or from DASD.

'20' New Page The buffer page is not from DASD; the buffer contains a new page.

'10' Data Space Page

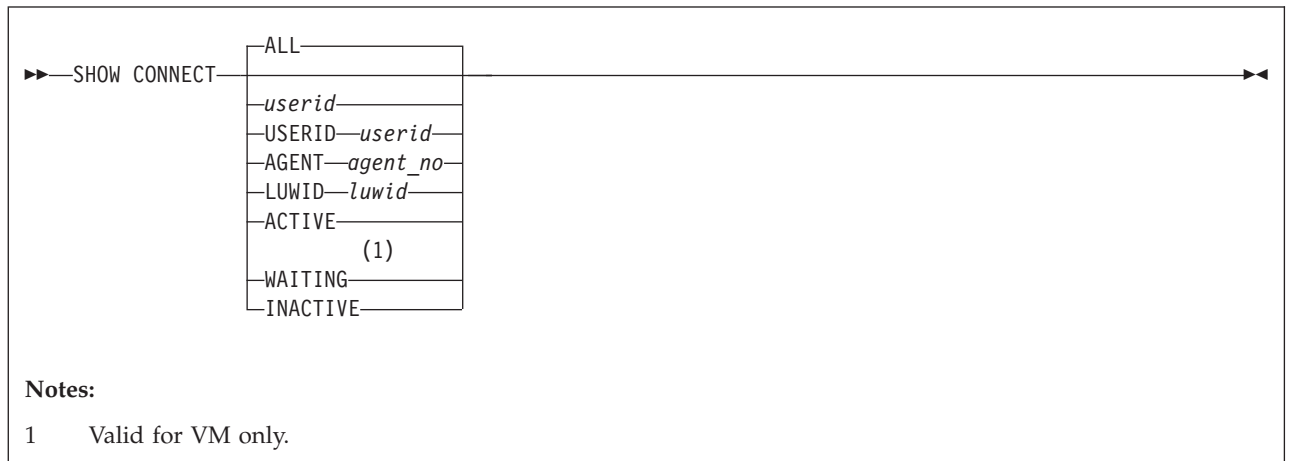
The buffer page is from a data space. This flag will only appear when the page in the buffer belongs to a storage pool that is using data spaces. For more information see the *DB2 Server for VSE & VM Performance Tuning Handbook* manual.

'00' No Flag No flag is present.

The above conditions can occur in combination and will result in the ORing of the individual bit values to display a FLAGS value for the combination. For example, you could have a modified page that is in transit back to DASD. This would combine the '80' and '40' bits to display a FLAGS value of 'C0'.

FIX CNT is the count of concurrent users of the page or directory block.

SHOW CONNECT



The `SHOW CONNECT` command displays the status information of all users or selected users connected to the application server.

The parameters of the `SHOW CONNECT` command have the following meanings:

ALL gives you information on all active, waiting, and inactive users connected to the application server.

USERID *userid*

is the DB2 Server for VSE & VM user ID of the person who is currently connected to the application server. This parameter limits the display to only the connected users with the user ID of *userid*. It is not necessary to specify the keyword `USERID`.

AGENT *agent_no*

is the agent number. This parameter limits the display to only the active agent specified in *agent_no*.

LUWID *luwid*

is the LUWID or some portion of the LUWID starting at the beginning up to the full length of the LUWID, which is 35 characters. This parameter limits the display to only the connected users for whom the equivalent portion of the LUWID matches the specified *luwid*.

ACTIVE

limits the display to all ACTIVE users.

WAITING (Valid in VM only)

limits the display to all WAITING users.

INACTIVE

limits the display to all INACTIVE users.

Note: The SHOW CONNECT ACTIVE command should be used before issuing the FORCE or FORCE RMTUSER commands to force users and remote users to end their work. The SHOW CONNECT INACTIVE command should be used before issuing the FORCE INACTIVE command to force inactive users to release their connection.

The following examples show the output of the SHOW CONNECT command.


```

MSG F4
AR 015 1I40I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show connect
F4 004 Status of Connected Users                1998-04-07 16:06:31
F4 004 Checkpoint agent is not active.
F4 004 User Agent: 1 User-ID: SAMUEL SQL-ID: SAMUEL
F4 004 is R/O APPL 12BCF
F4 004 Agent is processing and is in communication wait.
F4 004 State started: 1994-12-02 15:21:22
F4 004 Conversation started: 1996-12-02 15:21:22
F4 004 Task No.: 147 RMID: 32 Term. id: 077D
F4 004 CPU time: 00:00:02
F4 004 Protocol: SQLDS via SNA
F4 004 User Agent: 2 User-ID: MARISSA SQL-ID: MARISSA
F4 004 is R/W APPL 12FC4
F4 004 Agent is processing and is in communication wait.
F4 004 State started: 1996-12-02 15:23:17
F4 004 Conversation started: 1996-12-02 15:23:15
F4 004 CPU time: 00:00:01
F4 004 LUWID: CAIBMOML.OECGW001.A6773D6F8611.0001
F4 004 EXTNAM: MARISSA.1
F4 004 Requester: SQLDS/VM V3.3.0 at TOIVMLB6
F4 004 Rmtuser ID: 2
F4 004 LU name: OMPGW001
F4 004 Task no.: 0000134
F4 004 Protocol: SQLDS via SNA
F4 004 User Agent: 3 User-ID: LAURA SQL-ID: LAURA
F4 004 is R/O APPL 12FC6
F4 004 Agent is processing and is in I/O wait.
F4 004 State started: 1994-12-02 15:23:24
F4 004 Conversation started: 1994-12-02 15:23:15
F4 004 CPU time: 00:00:01
F4 004 LUWID: CAIBMOML.OECGW001.A6773D6E52C9.0001
F4 004 EXTNAM: LAURA.1
F4 004 Requester: SQLDS/VM V3.3.0 at TOIVMLB6
F4 004 Rmtuser ID: 3
F4 004 LU name: OMPGW001
F4 004 Task no.: 0000133
F4 004 Protocol: SQLDS via SNA
F4 004 User ID: ANDREW SQL ID: ANDREW
F4 004 User is inactive.
F4 004 State started: 1994-12-02 14:58:18
F4 004 Conversation started: 1994-12-02 14:58:16
F4 004 CPU time: 00:00:00
F4 004 LUWID: CAIBMOML.OECGW001.A67737DC6BEB.0001
F4 004 EXTNAM: ANDREW.1
F4 004 Requester: SQLDS/VM V3.3.0 at TOIVMLB6
F4 004 Rmtuser ID: 1
F4 004 LU name: OMPGW001
F4 004 Task no.: 0000110
F4 004 Protocol: SQLDS via SNA
F4 004 User ID: OLIVER SQL ID: OLIVER
F4 004 User is inactive.
F4 004 State started: 1999-03-15 10:56:29
F4 004 Conversation started: 1999-03-15 10:56:28
F4 004 CPU time: 00:00:00
F4 004 LUWID: G9151F6D.AE07.02DB15155638.0000
F4 004 EXTNAM: db2bp.exe 324851734 015E017D
F4 004 Requester: DB2 Connect 5.2 (DB2/NT) at BEDROCK
F4 004 Protocol: DRDA via TCP/IP
F4 004 3 remote users are active.
F4 004 2 remote users are inactive.
F4 004 1 agents are available.
F4 004 1 remote connections are available.
F4 004 ARI0065I Operator command processing is complete.

```

Figure 38. A Typical SHOW CONNECT Display on a VSE Operator Console or ISQL Terminal

```

show connect
Status of Connected Users                               1996-02-04 10:25:33
Checkpoint agent is not active.
User Agent: 1 User-ID: SHUM SQL-ID: SHUM
  is R/W APPL 7B1
  Agent is processing with LPAGEBUFF=1032
    State started: 1996-02-04 10:15:30
    Conversation started: 1996-02-04 10:12:45
    CPU time: 00:00:01
    Protocol: SQLDS via SNA
    Connection Id: 6
User Agent: 2 User-ID: SQLUSRSS SQL-ID: SQLUSRSS
  is R/O APPL 30BD
  Agent is not processing and is in communication wait.
    State started: 1996-02-04 09:48:28
    Conversation started: 1996-02-04 09:48:00
    CPU time: 00:00:02
    LUWID: CAIBMOML.*IDENT.A532D460755B.0001
    EXTNAM: SQLUSRSS.1
    Requester: SQLDS/VM V3.3.0 at TORVMLB4
    Protocol: SQLDS via SNA
    Connection Id: 7
User Agent: 3 User-ID: PETERSON SQL-ID: PETERSON
  is R/O APPL 3758
  Agent is processing and is in LOCK wait.
    State started: 1996-02-04 10:23:11
    Conversation started: 1996-02-04 10:22:15
    CPU time: 00:00:01
    LUWID: CAIBMOML.STLLU.A5241A50FABD.0001
    EXTNAM: PETERSON .BATCH .PETERSON.DSNESPRR
    Requester: DB2 V2.3.0 at IBMREGRDBSTL0012
    Protocol: DRDA via SNA
    Connection Id: 9
User Agent: 4 User-ID: VSEMCH10 SQL-ID: SQLDBA
  is R/O SUBS 1796
  Agent is not processing and is in communication wait.
    State started: 1996-09-08 10:42:55
    Conversation started: 1996-09-08 10:42:43
    Task no.: 371 RMID: 12 Term. id: N/A
    Connection Id: 16
User-ID: SWAGRMAN SQL-ID: SQLDBA
  User is waiting for an agent
    State started: 1992-02-04 10:22:11
    Conversation started: 1992-02-04 10:03:05
    CPU time: 00:00:02
    LUWID: IBMNET07.*IDENT.AB457DFF69BC.0001
    EXTNAM: SWAGRMAN.1
    Requester: SQLDS/VM V3.3.0 at TOROLAB3
    Protocol: SQLDS via SNA
    Connection Id: 13
User-ID: JAVIER SQL-ID: JAVIER
  User is inactive
    State started: 1992-02-04 10:02:11
    Conversation started: 1992-02-04 09:27:49
    CPU time: 00:00:03
    LUWID: IBMNET07.*IDENT.AB457DFF6ABC.0001
    EXTNAM: JAVIER.1
    Requester: SQLDS/VM V3.3.0 at TOROLAB
    Protocol: SQLDS via SNA
    Connection Id: 62

4 users are active.
1 users are waiting.
1 users are inactive.
0 agents are available.
94 user connections are available.
ARI0065I Operator command processing is complete.

```

Figure 39. A Typical SHOW CONNECT Display on a VM Operator Console or ISQL Terminal

The SHOW CONNECT output values have these meanings:

Header line

contains a timestamp that identifies when this status is taken.

Agent Status

contains the status of each active agent structure. Refer to “SHOW ACTIVE” on page 65 for information.

State started

is the timestamp when the current state began.

Conversation started

is the timestamp when the user was connected to the application server.

CPU time

is the CPU time since the user was connected to the application server. It is only displayed if you start the application server with accounting on (ACCOUNT=D or ACCOUNT=T in VSE).

LUWID

is a unique identifier associated with each connection. It is composed of four parts. For SNA connections, they are: network id, LU name, LUW instance number, and LUW sequence number. For TCP/IP connections, they are: IP address, port number, LUW instance number, and LUW sequence number. For VM/ESA local and TSAF connections, the LUWID will have a real LU name. For VSE/ESA, SNA remote connections, it is the LUNAME (APPLID) of the local CICS subsystem through which a remote user can be uniquely identified.

EXTNAM

is the name of the job that is being serviced by the application requester. If the LUW is initiated from a VM application requester, the external name is composed of the user’s logon ID and the current CMS work unit number.

Requester

is the identifier of the requesting system. This field is limited to nine characters.

Protocol *protocol via connection*

is the protocol that the requester is using.

SQLDS the non-DRDA (private) protocol is being used.

DRDA the DRDA protocol is being used with no sync point manager involvement. This means that the LUW is only capable of multiple-site read, single-site update activity.

DRDA2 Phase Commit

the DRDA protocol is being used with sync point manager involvement. This means that the LUW is capable of multiple-site read, multiple-site update activity.

with the following connection:

SNA the specified protocol is connected via SNA.

TCP/IP the specified protocol is connected via TCP/IP.

Package is the name of the package being executed. If the agent is not processing, this is the last package executed. This line is only displayed if the work status is R/O or R/W.

Section is the number of the section in the package being executed. If the agent is not processing, this is the last section

executed. This line is not displayed when the application is using modifiable extended dynamic SQL.

RMID is the Resource Manager ID specified by the CIRB transaction in VSE, and for VSE guest sharing users in VM. This identifies which resource adapter is used by the agent, because multiple resource adapters may exist. The RMID can be used, along with the CICS task number, to identify which agent should be forced before terminating a CICS transaction.

Term. ID is the CICS terminal ID. This identifies which CICS transaction is accessed by the CICS user. The terminal ID can be used, along with the CICS task number and the RMID, to identify which agent should be forced before terminating a CICS transaction. The value 'N/A' indicating that the terminal ID is not available, is displayed if there is no terminal ID associated with the agent.

Connection ID (Valid in VM only)

The connection identifier. This identification number can be used in the FORCE INACTIVE operator command to force a particular inactive user to release their connection.

Rmtuser ID (Valid in VSE only)

The remote user identifier. This identification number can be used in the FORCE RMTUSER operator command to force a particular remote user to end their work.

Task No. This is the CICS task number for CICS users, or the CICS task number of a connected APPC-to-XPCC Exchange Transaction (AXE) for remote (DRDA) users. For CICS users, the task number can be used to identify which agent should be forced before terminating the CICS transaction. For remote users, the task number, together with the LU Name, can be used to stop a particular remote user-initiated AXE transaction by using the CICS CEMT transaction, thereby forcing a specific remote user to end his work and end the conversation. This field is not displayed if there are no CICS tasks associated with the agent (for example, when an agent is not in work (NIW)).

Note: The LUWID, EXTNAM, and Requester lines are not displayed in the following situations:

1. The agent represents a VSE local user.
2. The agent represents a VSE guest in VM.
3. The application server and the application requester have not yet exchanged identification. In VSE, the Application Requester (AR) server class, the AR external name, and possibly the LUWID, may not yet be known to the Application Server.
4. The agent does not use Distributed Relational Database Architecture (DRDA) protocol.
5. The following is valid in a VSE environment only:

The State started and the Conversation started timestamps may be later than the timestamp in the header line. This occurs because of the multiple tasks that are handled by the application

server. Conversations can be started and states changed while the SHOW CONNECT command is processing.

SHOW CONNECT Enhancements for CRR in VM/ESA

When DB2 Server for VM is performing *resynchronization initialization* with the CRR recovery server, its status may be seen by the SHOW CONNECT. The following message is displayed:

```
Recovery Agent is processing Resynchronization Initialization
```

If the database manager is in a communications wait, waiting for the CRR recovery server to reply to its exchange log names request, the following message is displayed:

```
Recovery Agent is processing Resynchronization Initialization  
and is in a communications wait with the CRR Recovery Server
```

When the CRR recovery server is performing *resynchronization recovery* with DB2 Server for VM, its status may be seen by the SHOW CONNECT. The following message is displayed:

```
Recovery Agent is processing Resynchronization Recovery
```

If the database manager is in a communications wait, waiting for the CRR recovery server to acknowledge its exchange log names and compare states replies, following message is displayed:

```
Recovery Agent is processing Resynchronization Recovery  
and is in a communications wait with the CRR Recovery Server
```

If the database manager is committing or rolling back the logical unit of work requested by the CRR recovery server, the following message is displayed:

```
Recovery Agent is processing Resynchronization Recovery  
and is waiting for a <commit|rollback> to complete.
```

This information is available in tokenized format ¹.

SHOW CONNECT Enhancements for Stored Procedures

If a stored procedure is executing when the command is invoked, the output shows the name of the stored procedure, the name of the module or phase associated with it, the name of the stored procedure package, and the number of the section that is running. This information is displayed only if the command is issued from the operator console, or if the user issuing the command has DBA authority.

If an agent is waiting for a stored procedure server, this is indicated in the output displayed for that agent.

If an agent is executing a stored procedure when the SHOW CONNECT command is invoked, any package information displayed will be for the stored procedure and not the package that issued the CALL statement. Following is an example of the output you might receive.

1. For more information about tokenized format, see "Appendix A" of the *DB2 Server for VSE & VM Diagnosis Guide and Reference manual*

```

show connect
Status of Connected Users                               1997-09-30 08:56:42
Checkpoint agent is not active.
User Agent: 1 User-ID: SQLUSRKJ SQL-ID: SQLUSRKJ
  is R/O APPL 1666
  Agent is not processing and is in communication wait.
  State started: 1997-09-30 08:56:39
  Conversation started: 1997-09-30 08:56:12
  Protocol: SQLDS
  Package: SQLDBA.MAINPGM Section: 4
  Procedure: PROC1 Modname: MYPROC
  Procedure Package: SQLDBA.MYPROC Section: 4
User Agent: 2 User-ID: SQLUSRJR SQL-ID: SQLUSRJR
  is R/O APPL 1667
  Agent is not processing and is waiting for a stored procedure
  server in group GROUP1
  State started: 1997-09-30 08:56:39
  Conversation started: 1997-09-30 08:56:12
  Protocol: SQLDS
  Package: SQLDBA.MAINPGM2 Section: 3
User Agent: 3 User-ID: SQLUSRTH SQL-ID: SQLUSRTH
  is R/O APPL 1668
  Agent is processing with LPAGEBUFF=1032
  State started: 1997-09-30 08:56:39
  Conversation started: 1997-09-30 08:56:12
  Protocol: SQLDS
  Package: SQLDBA.MAINPGM3 Section: 4
  Procedure: PROC3 Modname: MYPROC3
  Procedure Package: SQLDBA.MYPROC3 Section: 2
User Agent: 4 User-ID: SQLUSRJR SQL-ID: SQLUSRJL
  is R/O APPL 1669
  Agent is not processing and is waiting for stored procedure
  PROC4 AUTHID SQLUSRJL to be started
  State started: 1997-09-30 08:58:00
  Conversation started: 1997-09-30 08:57:35
  Protocol: SQLDS
  Package: SQLDBA.MAINPGM4 Section: 3
4 Users are active.
0 Users are waiting.
0 Users are inactive.
2 Agents are available.
2 User connections are available.

ARI0065I Operator command processing is complete.

```

Figure 40. A Typical SHOW CONNECT Display

SHOW CRR LOGNAMES (VM Only)

►—SHOW—CRR—LOGNAMES—◄

The SHOW CRR LOGNAMES command can be used when log name mismatch situations arise between the database manager and VM/CRR. When this command is issued, the database manager displays:

- The CRR recovery server's log name found in the DB2 Server for VSE & VM log. This log name can be up to 64 characters long.
- The CRR recovery server's TPN found in the DB2 Server for VSE & VM log.
- The database manager's log name found in its own log.
- The current TPN for the database manager.
- The database manager's current log status relative to the CRR recovery server. This can be warm or cold.

Because the CRR recovery server is local to the database manager, the LUNAME is *IDENT and is not displayed.

Sample output is as follows:

```
show crr lognames
SHOW CRR LOGNAMES entered at DATE='mm-dd-yyyy' TIME='hh:mm:ss'
Log Name information for the database manager:
  TPN:      tpn
  LOG NAME: log name
Log Name information for the CRR recovery server:
  TPN:      tpn
  LOG NAME: log name
At the last Resynchronization Initialization,
the log status was warm|cold.
ARI0065I Operator command processing is complete.
```

Figure 41. Sample output from SHOW CRR LOGNAMES command (SYNCPNT=Y)

Note: When SYNCPNT=N, this command may still be executed. Sample output is as follows:

```
show crr lognames
SHOW CRR LOGNAMES entered at DATE='mm-dd-yyyy' TIME='hh:mm:ss'
Log Name information for the database manager:
  TPN:      tpn
  LOG NAME: log name
Log Name information for the CRR recovery server:
  TPN:      tpn
  LOG NAME: log name
There is no log status since SYNCPNT = N
ARI0065I Operator command processing is complete.
```

Figure 42. Sample output from SHOW CRR LOGNAMES command (SYNCPNT=N)

SHOW DBCONFIG

```
▶▶—SHOW DBCONFIG—◀◀
```

The SHOW DBCONFIG command displays the database configuration chosen when the database was *originally* generated. It can help you to determine whether you can increase the size of the database, by displaying its potential maximum size.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show dbconfig
F4 004 Current system identification      = DB2 VSE & VM 7.2
F4 004 System identification at DB generation = DB2 VSE & VM 6.1
F4 004
F4 004 DBA specified the following at DB definition:
F4 004   Maximum pools =                32
F4 004   Maximum DBEXTENTS =            64
F4 004   Maximum DBSPACES =             1000
F4 004
F4 004 Computed:
F4 004   Total number of DBSPACE blocks =   3582
F4 004   Total amount of DBSPACE =         916992 K
F4 004   Total number of physical pages =  233472
F4 004   Total amount of physical space =  933888 K
F4 004   Total number of directory blocks =  3800
F4 004
F4 004   Number of DBSPACE blocks left =   400

```

Figure 43. A Typical SHOW DBCONFIG Display on an Operator Console

```

show dbconfig
Current system identification      = DB2 VSE & VM 7.2
System identification at DB generation = DB2 VSE & VM 6.1

DBA specified the following at DB definition:
Maximum pools =                32
Maximum DBEXTENTS =            64
Maximum DBSPACES =             1000

Computed:
Total number of DBSPACE blocks =   3582
Total amount of DBSPACE =         916992 K
Total number of physical pages =  233472
Total amount of physical space =  933888 K
Total number of directory blocks =  3800

Number of DBSPACE blocks left =   400

```

Figure 44. A Typical SHOW DBCONFIG Display on an Operator Console or ISQL Terminal

SHOW DBCONFIG displays this information:

Current system identification

The current version and release level of the database manager. This number changes after you migrate to a newer version.

System identification at DB generation

The DB2 Server for VSE & VM version number when the database was *originally* generated. This version number does not change if you migrate to a newer version.

Maximum pools

Maximum number of storage pools. A storage pool is made up of one or more database extents (DBEXTENTS).

Maximum DBEXTENTS

Maximum number of database extents (dbextents). Dbextents are the physical media in which the database manager stores the data. Internally, all dbextents appear like a continuous DASD space of physical 4096-byte (4-kilobyte) pages.

It is possible to reach the maximum physical page limit before reaching the maximum number of dbextents. For more information, see "Total number of physical pages" below. For information on what to do if you reach the maximum physical page limit, refer to the *DB2 Server for VSE & VM Database Administration*.

Maximum DBSPACES

This is the maximum number of dbspaces (includes public, private, and internal dbspaces). The dbspaces represent a user-defined amount of continuous address space. (A user works with dbspaces; the application server maps dbspaces to physical dbextents.)

These values, which represent the potential database size, are also shown:

Total number of DBSPACE blocks

This value is the total number of database directory blocks minus the number of blocks used for system control blocks. This number includes both dspace blocks that are already allocated, and dspace blocks available for adding new dbspaces. The dspace blocks map the logical pages of a dspace to the physical pages of a dbextents in the storage pool to which the dspace is assigned. See "Number of dspace blocks left", below, for details on dspace blocks required for dbspaces.

Total amount of DBSPACE

This is the maximum amount, in kilobytes, of space that you can allocate to dbspaces in the database.

TOTAL AMOUNT OF DBSPACE = total number of dspace blocks * 256K
--

Total number of physical pages

This value, the maximum physical size of the database, is the maximum amount of dbextent space (in 4-kilobyte pages) that you can allocate to the database.

Total amount of physical space

This value is the total number of physical pages multiplied by 4 kilobytes.

Total number of directory blocks

This is the total number of 512-byte blocks in the database directory.

Number of DBSPACE blocks left

The number of dspace blocks left determines the amount of dspace that you can add to the database. This is the total number of blocks not yet allocated to a dspace, and available for future ADD DBSPACE functions. Every 128 pages of public or private dspace defined requires two dspace blocks. Every 128 pages of internal dspace defined requires one dspace block.

SHOW DBEXTENT

►—SHOW DBEXTENT—◄

A dbextent is the physical medium where the database manager stores data. Internally, all database extents appear like a continuous DASD address space of physical 4096-byte (4-kilobyte) pages. A storage pool is made up of one or more database extents.

The SHOW DBEXTENT command displays physical storage information about each storage pool defined. You can use this command to monitor physical storage usage. The SHOW DBEXTENT command gives output identical to the SHOW POOL command with the SUMMARY parameter. For a more detailed display of the dbextents and pools, refer to “SHOW POOL” on page 118.

The following examples show the output of the SHOW DBEXTENT command.

```
AR
MSG F4
AR 015 1I40I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show dbextent
F4 004 POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
F4 004 NO.      PAGES      PAGES USED  FREE PAGES  RESV PAGES  USED  EXTENTS  SOS
F4 004 1        741        178         563         20          24     1
F4 004 2        171        11          160         20          6      1
F4 004 3        114        0           114         20          0      1
F4 004 4        114        0           114         20          0      1
F4 004 FREE    22500
```

Figure 45. A Typical SHOW DBEXTENT Display on a VSE Operator Console

```
show dbextent
POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
NO.      PAGES      PAGES USED  FREE PAGES  RESV PAGES  USED  EXTENTS  SOS
1        741        178         563         20          24     1
2        171        11          160         20          6      1
3        114        0           114         20          0      1
4        114        0           114         20          0      1
FREE    22500
```

Figure 46. A Typical SHOW DBEXTENT Display on a VM Operator Console or ISQL Terminal

The columns of the output of SHOW DBEXTENT have these meanings:

POOL NO.

is the number of the storage pool where the dbextents reside. If the value of the number is positive, the database manager does recovery for the storage pool through logging all changes to the pool. (This is the usual situation.) The database manager records all changes to a recoverable pool. If the value is negative, the storage pool is nonrecoverable. The database manager does not log changes to a nonrecoverable pool. The database manager does however, take a checkpoint after every LUW; therefore, partial recovery is still available. The absolute value of a negative storage pool number is the number of the storage pool. For example, -54 indicates that storage pool number 54 is nonrecoverable.

TOTAL PAGES

is the total number of pages in the storage pool.

NO. OF PAGES USED

is the number of pages within the storage pool that have been acquired for use as data, header, or index pages. This value is the actual number of physical pages acquired. The number of logical pages required in the database is usually less than this value.

NO. OF FREE PAGES

is the number of pages within the storage pool that are currently not in use. The value represents the difference between TOTAL PAGES and NO. OF PAGES USED.

NO. OF RESV PAGES

is the number of reserved pages in the storage pool. A minimum of 20 pages is always reserved. See the *DB2 Server for VSE & VM Diagnosis Guide and Reference* for a description of reserved pages.

% USED

is the percentage of pages in the storage pool that are currently being used.

NO. OF EXTENTS

is the number of dbextents defined for that storage pool.

SOS

is the short on storage column. An asterisk (*) appears in this column when the storage pool is running out of storage (otherwise the column is blank). The level that sets this flag is the value you used for the SOSLEVEL parameter when you started the application server.

FREE

is the amount of free space in the database directory available for physical storage.

SHOW DBSPACE

```
▶—SHOW DBSPACE—n—▶
```

The database manager stores data in logical collections called dbspaces. A dbspace consists of multiples of 128 pages of linear space. There are three types of pages in a dbspace: header, data, and index.

Header pages contain control information about the contents of the dbspace. Data pages contain the tables created in the dbspace. Index pages contain the indexes on tables created in the dbspace. You choose the number of pages for each type when you acquire the dbspace. After you specify these numbers, you cannot change them.

SHOW DBSPACE lets you monitor the number of pages used by any public or private dbspace. In the SHOW DBSPACE command, *n* is the number of the dbspace you want to show. The dbspace number is in the DBSPACENO column of the SYSDBSACES catalog. Note that dbspace 1 is the catalog dbspace. The following is an example of the SQL SELECT statement for obtaining a dbspace number:

```
SELECT DBSPACENO FROM SYSTEM.SYSDBSPPACES
WHERE DBSPACENAME = <name of desired dbspace>
```

You can enter this SELECT statement from the ISQL facility or from the DBS Utility.

The SHOW DBSPACE command may take a long time to be completed if the dbspace is large.

The columns of output resulting from a SHOW DBSPACE command have the following meanings:

TYPE OF PAGES

is the type of page: header, data, or index.

NUMBER OF PAGES

is the total number of pages for each type.

NUMBER OF OCCUPIED PAGES

is the number and percentage of currently occupied pages for each type.

% FREE SPACE

is the percentage of free space that exists in currently occupied pages.

NUMBER OF EMPTY PAGES

is the number of pages that are allocated to the dbspace that contain no data. If the field has a large value, you may want to reorganize the dbspace to release the empty pages back to the storage pool. You can also use the SQLRELEP EXEC in VM, or start the database with STARTUP=P in VSE. See the *DB2 Server for VSE & VM Database Administration* for information on reorganizing dbspaces and releasing empty pages.

The following examples show the output of the SHOW DBSPACE command.

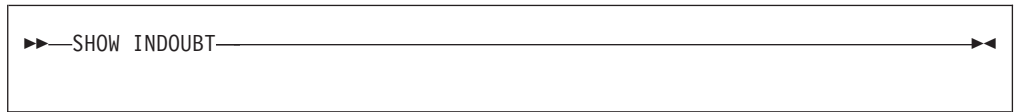
```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show dbspace 3
F4 004 TYPE      NUMBER      NUMBER OF      % FREE      NUMBER OF
F4 004 OF PAGES  OF PAGES    OCCUPIED PAGES SPACE      EMPTY PAGES
F4 004 HEADER      8          1 ( 12 %)    73 %        0
F4 004 DATA     1365       0 ( 0 %)     *           0
F4 004 INDEX      128        27 ( 21 %)   79 %        18
F4 004 ARI0065I Operator command processing is complete.
```

Figure 47. A Typical SHOW DBSPACE Display on a VSE Operator Console

```
show dbspace 3
TYPE      NUMBER      NUMBER OF      % FREE      NUMBER OF
OF PAGES  OF PAGES    OCCUPIED PAGES SPACE      EMPTY PAGES
HEADER      8          1 ( 12 %)    73 %        0
DATA     1365       0 ( 0 %)     *           0
INDEX      128        27 ( 21 %)   79 %        18
ARI0065I Operator command processing is complete.
```

Figure 48. A Typical SHOW DBSPACE Display on a VM Operator Console or ISQL Terminal

SHOW INDOUBT



The SHOW INDOUBT command displays the heuristic decisions made on in-doubt LUWs. SHOW INDOUBT displays the TRANID, USERID, LUWID, EXTNAM, REQUESTER, PACKAGE, STATE, DAMAGE, Prepared-To-Commit state started timestamp, and Heuristic state started timestamp for all in-doubt LUWs involved in two phase commits. In a VSE system, SHOW INDOUBT also displays RECTOK information.

Note: Any guest sharing agents which have been forced will not be displayed in SHOW INDOUBT outputs.

The following are examples of the SHOW INDOUBT command.

```
show indoubt
  TRANID: 1 User ID: FRED
    is prepared for COMMIT or ROLLBACK.
    LUWID: CAIBMOML.OMXRCB0A.ACA86699EA6A.0001
    EXTNAM: DB2BP.EXE          00940001
    Requester: DDCS/2   V2.1.0   at BEDROCK
    Package: NULLID.SQLC24D0   Section: 201
    PTC State started: 1996-04-02  10:11:18
    Heuristic state started: N/A
    Damage: No
  TRANID: 2 User ID: BARNEY
    is COMMITTED-H.
    LUWID: CAIBMOML.OMXRCB0A.ACA86699EA6A.0002
    EXTNAM: DB2BP.EXE          00940001
    Requester: DDCS/2   V2.1.0   at BEDROCK
    Package: NULLID.SQLC24D0   Section: 201
    PTC State started: 1996-04-02  09:13:27
    Heuristic state started: 1996-04-02  09:52:03
    Damage: Yes
```

Figure 49. Sample output from SHOW INDOUBT command on VM

```

4 show indoubt
F4 004   TRANID: 1 User ID: WILMA
F4 004       is prepared for COMMIT or ROLLBACK.
F4 004       LUWID: CAIBMOML.OMXRCB0A.ACA86699EA6A.0001
F4 004       EXTNAM: DB2BP.EXE           00940001
F4 004       Requester: DDCS/2    V2.1.0   at BEDROCK
F4 004       Package: NULLID.SQLC24D0   Section: 201
F4 004       PTC State started: 1996-04-02 10:11:18
F4 004       Heuristic state started: N/A
F4 004       Damage: No
F4 004       RECTOK: X'AC89520F AB638400'
F4 004   TRANID: 2 User ID: BETTY
F4 004       is ROLLEDBACK-H.
F4 004       LUWID: CAIBMOML.OMXRCB0A.ACA86699EA6A.0002
F4 004       EXTNAM: DB2BP.EXE           00940001
F4 004       Requester: DDCS/2    V2.1.0   at BEDROCK
F4 004       Package: NULLID.SQLC24D0   Section: 201
F4 004       PTC State started: 1996-04-02 09:13:27
F4 004       Heuristic state started: 1996-04-02 09:52:03
F4 004       Damage: Yes
F4 004       RECTOK: X'AC935204 AB637432'

```

Figure 50. Sample output from SHOW INDOUBT command on VSE

SHOW INDOUBT displays one of the following three messages after the user ID for each in-doubt LUW:

- IS PREPARED FOR COMMIT OR ROLLBACK for a LUW that completed the first phase of the two phase commit.
- COMMITTED-H for a LUW that completed the first phase of the two phase commit and then a heuristic action was performed to commit the updates.
- ROLLEDBACK-H for a LUW that completed the first phase of the two phase commit and then a heuristic action was performed to rollback the updates.

TRANID

The DB2 Server for VSE & VM transaction ID. TRANID is initialized to 0 when COLDLOG is performed and will be incremented by 1 for each new LUW. TRANID is unique in DB2 Server for VSE & VM.

USERID

The DB2 Server for VSE & VM user ID of the person who was connected to the application server for the in-doubt LUW.

LUWID

The SNA LUWID.

EXTNAM

The name of the job (task, process, etc.) that is being serviced by the AR (same as the SHOW CONNECT output).

REQUESTER

The identifier of the requesting system from the srvnam field, the server class and version (same as the SHOW CONNECT output).

PACKAGE

The name of package being executed or the name of the last package executed (same as the SHOW CONNECT output).

SECTION

The section number of the package or the section number of the last package executed (same as the SHOW CONNECT output).

STATE

C(Commit) or B(Backout) or P(Prepare to Commit)

DAMAGE

Y(Heuristic damage has been done), N(Heuristic damage has not been done), or U(Unknown).

Prepare-To-Commit state started

Time from Prepare-To-Commit (PTC) log record done. This is a S/390 TOD clock value.

Heuristic state started

Time when FORCE was performed. This is a S/390 TOD clock value. Note that this field is displayed only when a heuristic action has been taken.

RECTOK

CICS recovery token. This is only used in the VSE environment.

SHOW INITPARM

▶—SHOW INITPARM—◀

The SHOW INITPARM command displays the current settings of each of the initialization parameters. This command can be used to decide whether fine-tuning of these parameters should be performed.

The following are examples of the SHOW INITPARM command.

```
System:  ARI0045I Ready for operator communications.
User:    show initparm

System:  Initialization parameters at 1996-02-27 15:29:27

        AMODE 24          DUALLOG N          PROTOCOL SQLDS
        SYNCNT N          ACCOUNT N          DBMODE G
        DUMPTYPE N        LOGMODE Y          STARTUP W
        SYSMODE M        EXTEND N           CHARNAME INTERNATIONAL
        DBNAME SQLMACH1   RESID SQLMACH1
        PARMID           TRACDBSS 000000000000 TRACDSC 00
        TRACRDS 0002100* TRACCONV 0          TRACSTG 1
        TRACEBUF 0        ARCHPCT 80         CHKINTVL 10
        NCSCANS 30        NCUSERS 5          NDIRBUF 30
        NLRBS 2520       NLRBU 1000         NPACKEGE 10
        NPACKEGE 30      NPAGBUF 30         SLOGCUSH 90
        SOSLEVEL 10      DISPBIAS 7         LTIMEOUT 0
        DSPSTATS 21      SECALVER N         SECTYPE DB2
        SERVICE SQLMACH1 TCPPORT 6100       IPADDR 9.33.33.33
        HOST thehost.system.ibm.com         PTIMEOUT 180
        PROCXAB 0        DB2LEVEL 7.2.0     TAPEMGR Y
        TCPMAXRT 158     TCPRETRY Y          ALTLOG N

        ARI0065I Operator command processing is complete.
```

Figure 51. SHOW INITPARM Display for VM

```

System:   F4-004 ARI0062A SQLDS :
          Enter a DB2 Server for VSE operator command.
User:    4 show initparm
System:  F4 004
          F4 004 Initialization parameters at 1996-02-27  20:30:20
          F4 004
          F4 004 DSPLYDEV B           DUALLOG N           SYNCPNT N
          F4 004 RMTUSERS 0           ACCOUNT N           DUMPTYPE F
          F4 004 LOGMODE Y           STARTUP W           SYSMODE M
          F4 004 EXTEND N            CHARNAME INTERNATIONAL
          F4 004 DBNAME SQLDS
          F4 004 PARMID              TRACDBSS 000000000000  APPLID SYSARI00
          F4 004 TRACRDS 00000000    TRACCONV 0           TRACSTG 0
          F4 004 TRACEBUF 0          ARCHPCT 80          CHKINTVL 10
          F4 004 NCSCANS 30          NCUSERS 5           NDIRBUF 30
          F4 004 NLRBS 2520          NLRBU 1000         NPACKAGE 10
          F4 004 NPACPCT 30          NPAGBUF 30         SLOGCUSH 90
          F4 004 SOSLEVEL 10         DISPBIAS 7          LTIMEOUT 0
          F4 004 DSPSTATS 21         PTIMEOUT 55        PROCMXAB 0
          F4 004 TCPPORT 8000        HOST VSEMCH01      IPADDR 9.21.23.104
          F4 004 DB2LEVEL 7.2.0      TAPEMGR Y          ARCHTAP REW
          F4 004 TCPMAXRT 158        TCPRETRY Y         ALTLOG N
          F4 004
          F4 004 ARI0065I Operator command processing is complete.

```

Figure 52. SHOW INITPARM Display for VSE

Notes:

1. This display is for example purposes only. Some parameters will not be displayed depending on whether, for instance, the command is processed on a VM or VSE server, or whether private or DRDA flows are being used.
2. Since tracing can be turned on for all agents or for just a subset of the agents, a way to distinguish between these two tracing methods is introduced for this display. If tracing is only turned on for a subset of the agents, the trace value is followed by the '*' character. If the tracing is for all agents, then no '*' will be displayed.

SHOW INVALID

```

▶▶—SHOW INVALID—◀◀

```

The SHOW INVALID command displays the entities in the application server that are invalid or transient. Only indexes can become invalid or transient.

An index can become invalid (unusable) if a system error occurs that affects the particular index, or if the application server ends abnormally during an operation involving an index and the system is restarted without a log.

An index becomes transient when a logical unit of work (LUW) is performing an operation that puts the index in a temporarily inconsistent state. The index becomes valid again after the operation or LUW is completed successfully. Some operations make an index transient for the operation only. Others make an index transient for the life of the LUW. A transient index should not cause concern, because the index will become valid during normal processing.

The columns of output from a SHOW INVALID command have the following meanings:

ENTITY

identifies an index as the entity.

DBSPACE NUMBER

is the number of the dbspace where the index resides.

Note: An index in the system catalog dbspace (SYS0001) cannot be marked invalid. It can however be marked transient. The application server will end if an attempt is made to invalidate an index in this dbspace.

ENTITY ID

is the internal identifier of the index. This corresponds to IID in the SYSINDEXES catalog. You can obtain the index name and owner from the SYSINDEXES catalog with this statement:

```
SELECT A.DBSPACENO, B.IID, B.INAME, B.CREATOR, A.TNAME
FROM SYSTEM.SYSCATALOG A, SYSTEM.SYSINDEXES B
WHERE A.TNAME = B.TNAME AND A.CREATOR = B.CREATOR
AND A.DBSPACENO IN (dbspace-no-list)
AND B.IID IN (index-no-list)
ORDER BY A.DBSPACENO, B.IID
```

where `dbspace-no-list` and `index-no-list` are lists of the invalid index IDs and their dbspace numbers.

DESCRIPTION

is the reason the entity is invalid or transient.

1. For invalid indexes:

No Room

An index is marked invalid during forward, ROLLBACK or UNDO processing when an insufficient number of physical or logical pages are available during an index creation or update. The index must be dropped and recreated or reorganized.

Incomplete Index because of CREATE or REORGANIZE

If a checkpoint is taken during a CREATE or REORGANIZE INDEX, a system failure occurs before the end of the statement, and the application server is restarted without a log, the index is marked invalid. In this case, the index can appear in the SHOW INVALID display but not in the catalog.

If an index appears in the SHOW INVALID display but does not appear in the catalog, the changes to the application server from a CREATE INDEX command were not recovered after a system failure.

If an index appears in the SHOW INVALID display and in the catalog but you cannot drop or reorganize it, the changes to the application server from a REORGANIZE INDEX command were not recovered after a system failure.

In both of these instances, either drop and recreate the dbspace containing the index, or restore a previous database archive.

Duplicates Exist in UNIQUE INDEX

A unique index can become invalid with duplicates in these cases:

- If a checkpoint is taken during a searched UPDATE that is deferring UNIQUE integrity checking, a system failure occurs before the end of the statement and the database manager is restarted without a log, a UNIQUE index might contain duplicates because the changes could not be rolled back.
- At the end of filtered log recovery, a unique index might contain duplicates because the combinations of bypass UNDO processing or rollback of committed work allowed duplicates to be inserted which were not resolved.
- If an attempt is made to create a unique index during recovery and duplicate data is encountered. This can occur while rolling back an LUW, which drops a unique index and inserts duplicate data into a table in a nonrecoverable storage pool. The nonrecoverable data inserts cannot be undone, and when the recoverable index drop is undone (by performing a CREATE INDEX), the duplicate data is detected. This can also occur when using filtered log recovery during warmstart or while applying logs during restore where duplicate values are introduced and a subsequent attempt is made to create a unique index.

To recover, the user drops the index, removes duplicates from the table, and recreates the index. After the index is dropped, the following statement displays the duplicate keys:

```
SELECT column-name FROM table-name -
GROUP BY column-name -
HAVING COUNT (*) > 1
```

where:

column-name

is the name of the index column

table-name

is the name of the table.

The user can then remove the duplicates by deleting or updating the rows. The user can then recreate the index. If the index was created in support of a primary key or unique constraint, the user must deactivate the primary key or unique constraint, remove the duplicates, and then activate the primary key or unique constraint again.

2. For transient indexes:

Incomplete Index because of CREATE or REORGANIZE

The index is incomplete because it is either being created or reorganized (using the DBSU REORGANIZE INDEX command). After the index is successfully created or reorganized, the index becomes valid again.

Searched UPDATE with Deferring

One or more LUWs are in progress that are using a unique index during a searched UPDATE where uniqueness checking is being deferred. This unique index might contain duplicates that will be resolved by the end of the LUWs.

Note that because an index can become transient for an operation or an LUW, an index can be transient for more than one reason.

The following examples show the output of the SHOW INVALID command.

```

show invalid

INVALID ENTITIES: 2
      DBSPACE  ENTITY
ENTITY  NUMBER  ID    DESCRIPTION
INDEX   15     -32342 No room
INDEX   11     -32849 Duplicates Exist in UNIQUE INDEX

TRANSIENT ENTITIES: 3
      DBSPACE  ENTITY
ENTITY  NUMBER  ID    DESCRIPTION
INDEX   13     -31816 Incomplete Index due to CREATE/REORGANIZE
INDEX   11     -33014 Searched UPDATE with deferring
INDEX   11     -32849 Incomplete Index due to CREATE/REORGANIZE
INDEX   11     -32849 Searched UPDATE with deferring

ARI0065I Operator command processing is complete.

```

Figure 53. A Typical SHOW INVALID Display on a VM Operator Console or ISQL Terminal

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show invalid
F4 004
F4 004 INVALID ENTITIES: 2
F4 004      DBSPACE  ENTITY
F4 004 ENTITY  NUMBER  ID    DESCRIPTION
F4 004 INDEX   15     -32342 No room
F4 004 INDEX   11     -32849 Duplicates Exist in UNIQUE INDEX
F4 004
F4 004 TRANSIENT ENTITIES: 3
F4 004      DBSPACE  ENTITY
F4 004 ENTITY  NUMBER  ID    DESCRIPTION
F4 004 INDEX   13     -31816 Incomplete Index due to CREATE/REORGANIZE
F4 004 INDEX   11     -33014 Searched UPDATE with deferring
F4 004 INDEX   11     -32849 Incomplete Index due to CREATE/REORGANIZE
F4 004 INDEX   11     -32849 Searched UPDATE with deferring
F4 004
F4 004 ARI0065I Operator command processing is complete.

```

Figure 54. A Typical SHOW INVALID Display on a VSE Operator Console

See the *DB2 Server for VSE & VM Diagnosis Guide and Reference* for more information about invalid and transient indexes.

SHOW LOCK ACTIVE

```

▶▶—SHOW LOCK ACTIVE—◀◀

```

The SHOW LOCK ACTIVE command displays locking information for agent structures. An agent structure is the internal representation of an active user (DB2 Server for VSE & VM application program, ISQL user, or system agent such as checkpoint). The following is shown for each agent structure:

AGENT

is the agent number (identifier).

Note that the agent identifier for the checkpoint agent is C.

USER is the DB2 Server for VSE & VM authorization ID of the person who is currently using the agent. If the agent is unused, it is represented with blanks. In VM, the authorization ID is typically the user's virtual machine user ID.

Note that the authorization ID of the checkpoint agent is CHECKPT.

If the database manager is in the process of verifying the authorization ID, it is not shown. The string *null id* is shown instead.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different authorization IDs, within separate CMS units of work with one work unit waiting for another. To see if a user is in this wait state, enter the SHOW USERS command. See Figure 83 on page 137 for information on what to do when a VM user is in a wait state for a resource or database under one or more authorization IDs.

WAIT STATE

can be:

CKPT if the agent is waiting for the application server to do a checkpoint of the database.

COMM

if the agent is in communication wait then it is waiting for an SQL request from the user, or is waiting for an APPC/VM function to end.

COMM

if the agent is waiting for an SQL request from the user. (That is, the agent is waiting for the user to ask it to do something.)

DSPF if the agent is waiting for a data space page fault to be resolved. ²

I/O if the agent is waiting for database input or output.

LOCK if the agent is waiting for a database resource.

NIW if the agent currently has no logical unit of work in progress (Not In Work). This agent has no locking activity.

OUTB if the agent is waiting for a block buffer (directory buffer).

OUTP if the agent is waiting for a page buffer.

SLD if the agent is waiting for a data space save list definition block. ²

TOTAL LOCKS

is the total number of locks the agent is holding.

LONG LOCKS

is the number of locks that the agent must continue to hold until the user ends the logical unit of work. (Some locks can be freed before the logical unit of work ends; LONG LOCKS cannot.)

² This wait state will only occur when the data spaces are actually being used. For more information, see "Checking the Status of Users" in the *DB2 Server for VSE & VM Performance Tuning Handbook* manual. SLD wait only applies to the checkpoint agent.

WANTLOCK TYPE

is the type of lock for which the agent is waiting. If the agent is not waiting for a lock, this field is blank. If the agent is waiting for a lock, the database manager indicates the type of lock with one of these terms:

DB for a lock on the database.

DBSP for a lock on an entire dbspace.

IKEY for a lock on a key of an index.

IPAG for a lock on a page of an index.

PAGE for a lock on a page (4096 bytes) of data.

ROW for a lock on a row of data.

SYS for a special internal system lock.

TABL for a lock on an entire table.

Note: A database lock is acquired with long duration in intention exclusive (IX) mode at the start of each logical unit of work. A special internal system lock can actually be a latch on a page or block buffer. This will show as a latch wait if the `SHOW ACTIVE` command is entered.

The database lock is for log archive checkpoints. The database manager always takes a checkpoint before creating a log archive or switching to the inactive log. The checkpoint agent acquires a database lock with short duration in exclusive (X) mode when taking the checkpoint. The checkpoint agent's acquisition of the database lock ensures that no logical units of work are active when the checkpoint that precedes the log archive is in progress. The checkpoint agent releases the database lock when the checkpoint ends.

For a lock type of `SYS`, you should use the `SHOW ACTIVE` command to display the status of the active agent structures.

WANTLOCK DBSPACE

is the number of the dbspace where the data to be locked resides. If `WANTLOCK TYPE` is `DB` or `SYS`, blanks appear in this column. dbspace number 1 contains the catalog tables. The number corresponds to the `DBSPACENO` column of `SYSTEM.SYSCATALOG` and `SYSTEM.SYSDBSPACES`. For example, to obtain more information about the dbspace that the number represents, use this statement:

```
SELECT * FROM SYSTEM.SYSDBSPACES
WHERE DBSPACENO = n
```

Specify the number of the dbspace for *n*.

The following examples show the output of the `SHOW LOCK ACTIVE` command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004 Enter a DB2 Server for VSE operator command.
4 show lock active
F4 004
F4 004 AGENT USER WAIT TOTAL LONG WANTLOCK WANTLOCK
F4 004 C CHECKPT NIW 0 0
F4 004 1 JESSICA I/O 140 100
F4 004 2 LEIGH LOCK 226 220 DBSP 12
F4 004 3 MARY COMM 97 49
F4 004 4 MIKE LOCK 108 101 DBSP 12
F4 004 5 BOB LOCK 107 98 PAGE 14
F4 004 6 IRENE LOCK 635 623 DBSP 14
F4 004 ARI0065I Operator command processing is complete.

```

Figure 55. A Typical SHOW LOCK ACTIVE Display on a VSE Operator Console

```

show lock active
AGENT USER WAIT TOTAL LONG WANTLOCK WANTLOCK
C CHECKPT NIW 0 0
1 JESSICA I/O 140 100
2 LEIGH LOCK 226 220 DBSP 12
3 MARY COMM 97 49
4 MIKE LOCK 108 101 DBSP 12
5 BOB LOCK 107 98 PAGE 14
6 IRENE LOCK 635 623 DBSP 14
ARI0065I Operator command processing is complete.

```

Figure 56. A Typical SHOW LOCK ACTIVE Display on a VM Operator Console or ISQL Terminal

SHOW LOCK DBSPACE

```

▶▶ SHOW LOCK DBSPACE ALL ◀◀
    └─┬──┘
        n

```

The SHOW LOCK DBSPACE command displays information about locks within a public or private dbspace. If you specify ALL, you see information for all public and private dbspaces that have lock activity. If you want information for a particular dbspace, specify the number of the dbspace for *n*. You must specify either ALL or *n*.

To display the number of a dbspace, query the DB2 Server for VSE & VM catalog SYSTEM.SYSDBSPACES. The DBSPACENO column of that table contains the dbspace number. For example, this SELECT statement finds the dbspace number for MYSPLACE:

```

SELECT DBSPACENO FROM SYSTEM.SYSDBSPACES
WHERE DBSPACENAME='MYSPLACE'

```

You can enter this SELECT statement from the ISQL facility or the DBS Utility. If dbspace number 1 is displayed, the catalog tables are locked.

For the dbspaces you specify in `SHOW LOCK DBSPACE`, the database manager displays a line for each lock type that is held within that dbspace. The following information is displayed in the output lines:

DBSPACE NUMBER

is the number of the dbspace for which information is being displayed. The number corresponds to the `DBSPACENO` column of `SYSTEM.SYSCATALOG` and `SYSTEM.SYSDBSPACES`.

LOCK TYPE

is the type of database resource that is locked. `LOCK TYPE` can be one of the following:

DBSP for a lock on an entire dbspace.

IKEY for a lock on a key of an index.

IPAG for a lock on a page of an index.

PAGE for a lock on a page (4 096 bytes) of data.

ROW for a lock on a row of data.

TABL for a lock on an entire table.

(One line is displayed for each active lock type within a dbspace.)

IN is the number of locks of the indicated type that are intention none locks.

SIX is the number of locks of the indicated type that are share and intention exclusive locks.

IS is the number of locks of the indicated type that are intention share locks.

IX is the number of locks of the indicated type that are intention exclusive locks.

S is the number of locks of the indicated type that are share locks.

U is the number of locks of the indicated type that are update locks.

X is the number of locks of the indicated type that are exclusive locks.

Z is the number of locks of the indicated type that are super exclusive locks.

For more information about lock modes, see the *DB2 Server for VSE & VM Diagnosis Guide and Reference* manual.

NUMBER WAITERS

is the number of users waiting for a lock of the type indicated.

The following examples show the output of the `SHOW LOCK DBSPACE` command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show lock dbspace 12
F4 004 DBSPACE LOCK
F4 004 NUMBER TYPE IN SIX IS IX S U X Z NUMBER
F4 004
F4 004 12 DBSP 0 0 0 3 0 0 0 0 0 0
F4 004 12 IKEY 0 0 0 3 38 0 0 0 0 0
F4 004 12 ROW 0 0 0 3 16 0 0 0 0 0
F4 004 12 TABL 0 0 0 3 0 0 0 0 0 0
F4 004 12 DBSP 0 0 0 3 0 0 0 0 0 0
F4 004 12 IPAG 0 0 0 3 0 0 0 0 0 0
F4 004 12 PAGE 0 2 0 4 0 1 2 0 0 1
F4 004 12 TABL 0 0 2 0 0 0 0 0 0 0
F4 004 ARI0065I Operator command processing is complete.

```

Figure 57. A Typical SHOW LOCK DBSPACE Display on a VSE Operator Console

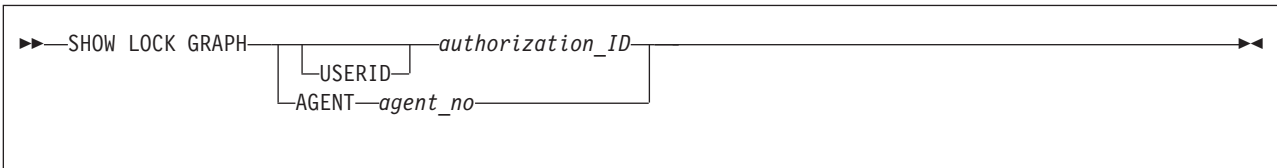
```

show lock dbspace 7
DBSPACE LOCK
NUMBER TYPE IN SIX IS IX S U X Z NUMBER
7 DBSP 1 0 0 2 0 0 0 0 0 0
7 IKEY 0 0 0 0 0 0 2 0 0 0
7 ROW 0 0 0 0 0 0 1 0 0 0
7 TABL 1 0 0 1 0 0 0 0 0 1
ARI0065I Operator command processing is complete.

```

Figure 58. A Typical SHOW LOCK DBSPACE Display on a VM Operator Console or ISQL Terminal

SHOW LOCK GRAPH



The SHOW LOCK GRAPH command displays information about the lock dependencies among agents. For the authorization ID (or agent number) you choose, SHOW LOCK GRAPH shows the users that must relinquish their locks before the specified user can get the lock.

Note: If you enter a nonunique authorization ID, the SHOW LOCK GRAPH command displays information about the lowest numbered agent that is processing for the authorization ID. This may not be the agent you are interested in. If not, enter a SHOW ACTIVE command to obtain the agent number. Then reenter the SHOW LOCK GRAPH command with the agent number.

You can think of the SHOW LOCK GRAPH command as displaying a hierarchy of waiting users. For example, if you enter SHOW LOCK GRAPH AGENT 2, the display might show that agent 2 is waiting for agents 5 and 7, and that agent 7 is waiting for agents 1, 4, and 6.

As shown in the above command format, you must specify either an authorization ID or an agent number. If you specify an agent number, you must enter the AGENT keyword. The USERID keyword is optional when you specify an authorization ID.

To get information about the checkpoint agent, specify either the letter C for the agent number or CHECKPT for the authorization ID.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different authorization IDs, within separate CMS units of work with one work unit waiting for another. To see if a user is in this wait state, enter the SHOW USERS command. See Figure 83 on page 137 for information on what to do when a VM user is in a wait state for a resource or database under one or more authorization IDs.

SHOW LOCK GRAPH displays the following:

LOCK REQUESTER

is the agent number followed by the authorization ID of the user who is currently using that agent. This agent is waiting for a lock held by the LOCK HOLDER.

LOCK HOLDER

is the agent number followed by the authorization ID of the user who is currently using that agent. This agent is preventing the LOCK REQUESTER from getting a lock. The LOCK HOLDER may be waiting for some other user to free a lock.

WAIT STAT (Wait State)

indicates the wait state of the agent of the LOCK HOLDER. The WAIT STAT can be:

- | | |
|-------------|--|
| CKPT | if the agent is waiting for the database manager to do a checkpoint for the database. |
| COMM | if the agent is waiting for an SQL request from the user. (That is, the agent is waiting for the user to ask it to do something.)

or, in VM, if the agent is in communication wait, it is waiting for an SQL request from the user or for an APPC/VM function to end. |
| DSPF | if the agent is waiting for a data space page fault to be resolved. ³ |
| I/O | if the agent is waiting for database input or output. |
| LOCK | if the agent is waiting for a database resource to become free. |
| OUTB | if the agent is waiting for a block buffer (directory buffer). |
| OUTP | if the agent is waiting for a page buffer. |
| SLD | if the agent is waiting for a data space save list definition block. ³ |

3. This wait state will only occur when the data spaces are actually being used. For more information, see "Checking the Status of Users" in the *DB2 Server for VSE & VM Performance Tuning Handbook* manual.

LOCK TYPE

is the type of database resource for which the LOCK REQUESTER has requested a lock. The LOCK TYPE can be:

- DB** for a lock on the database.
- DBSP** for a lock on an entire dbspace.
- IKEY** for a lock on a key of an index.
- IPAG** for a lock on a page of an index.
- PAGE** for a lock on a page (4 096 bytes) of data.
- ROW** for a lock on a row of data.
- SYS** for a special internal system lock.
- TABL** for a lock on an entire table.

Note: A database lock is acquired with long duration in intention exclusive (IX) mode at the start of every logical unit of work. A special internal system lock can actually be a latch on a page or block buffer. This will show as a LATCH WAIT if the SHOW ACTIVE command is entered.

The database lock is for log archive checkpoints. The database manager always takes a checkpoint before creating a log archive or switching to an alternate log. When the database manager takes a checkpoint, the checkpoint agent acquires a database lock with short duration in exclusive (X) mode. The checkpoint agent's acquisition of the database lock ensures that no logical units of work are active when the checkpoint that precedes the log archive is in progress. The checkpoint agent releases the database lock at the end of the checkpoint.

DBSP NUMBR (Dbspace Number)

is the number of the dbspace where the LOCK REQUESTER requested a lock. If the LOCK TYPE is DB or SYS, blanks appear in this column. The values for DBSP NUMBR correspond to those in the DBSPACENO columns of SYSTEM.SYSCATALOG and SYSTEM.SYSDBSPACES.

LOCK QUALF

is an entity that the database manager uses internally to qualify the lock that is wanted. The LOCK QUALF varies depending on LOCK TYPE:

LOCK TYPE	LOCK QUALF
DB	blanks.
DBSP	blanks, because the dbspace number (which is the qualifier) is already displayed under DBSP NUMBR.
PAGE	the page number that the database manager previously assigned to the page.
TABL	the internal identifier that the database manager uses for the table. The values in TABL correspond to those in the TABID column of SYSTEM.SYSCATALOG.
ROW	the internal identifier that the database manager uses for the row.
IKEY	an internal value that the database manager uses to represent the index key. The Lock Qualifier can be an

internal ROW identifier and an internal INDEX identifier if the index is unique and KEY LEVEL Locking is in effect.

IPAG the index page number. The database manager assigns numbers to pages that contain indexes in the same manner that it assigns data page numbers.

SYS blanks.

The values PAGE, ROW, IKEY, and IPAG are useful only for pattern matching during problem determination. The DBSP and TABL values correspond to the DBSPACENO and TABID columns in the DB2 Server for VSE & VM catalogs. For example, the following SELECT statement identifies the table by using the DBSP and TABL values:

```
SELECT TABID, TNAME FROM SYSTEM.SYSCATALOG
WHERE TABID=tabl-value AND DBSPACENO=dbsp-value
```

REQ STATE (Request State)

is the status of the lock that the LOCK REQUESTER requested. The REQ STATE can be:

C WAIT

which means waiting to be converted. The agent has made a lock request, but the lock cannot be allocated now. The agent had already locked a database resource in a particular lock mode. Now the agent wants to change the mode of the lock it acquired, so it issues a new lock request. The agent could issue a new lock request when, for example, a user is doing read-only work during a logical unit of work, and then wants to modify the data. The database manager needs to change a read-only lock to a write lock. This is a lock conversion.

If some other user also held a lock on the resource, the requested mode could be incompatible. A conversion is not immediately possible. The user waits because the lock is waiting to be converted.

G WAIT

which means waiting to be granted. The agent made a lock request, but someone else has already locked the resource. The agent is waiting to be granted the lock.

DENIED

which means a requested lock was not available, and the database manager canceled the request. That is, the lock was denied and the agent is not waiting for it.

In this situation, the agent requested a lock that was not available. The agent indicated by its request that, if no lock was available, it would not wait for the lock. The database manager denied the request rather than put the agent in a lock wait. An example of this kind of request is a user trying to access a private dbspace that is currently in use. If the private dbspace is currently in use and the lock request is not compatible, the database manager does not let the lock request wait. Instead, the database manager denies the request.

REQ MODE (Request Mode)

is the mode of the lock that the agent waiter requested:

- IN** indicates that the waiter requested an intention none lock.
- SIX** indicates that the waiter requested a share and intention exclusive lock.
- IS** indicates that the waiter requested an intention share lock.
- IX** indicates that the waiter requested an intention exclusive lock.
- S** indicates that the waiter requested a share lock.
- U** indicates that the waiter requested an update lock.
- X** indicates that the waiter requested an exclusive lock.
- Z** indicates that the waiter requested a super exclusive lock.

For more information about lock modes, see the *DB2 Server for VSE & VM Diagnosis Guide and Reference*.

DUR (Duration)

indicates the duration of the lock that the waiter requested. DUR can be:

- INST (instant)** used to test the request state of the lock. That is, to test if a particular database resource (an index key, for example) is in use. The lock will be freed when the request is granted.
- LONG** indicates that the lock, when granted, is held until the end of the logical unit of work.
- MED (medium)** indicates that the lock, when granted, will be held until the database manager explicitly releases it, or until the end of the logical unit of work.
- SHORT** indicates that the waiter will not require this lock for the entire duration of the logical unit of work. Instead, the lock will be released at the end of the call to the Database Storage Subsystem (DBSS).

The following figures show that LEIGH (the specified user) is waiting for MARK. LEIGH wants a long lock on the entire dbspace 15. The WAIT STAT column on that first line shows that MARK (the LOCK HOLDER) is in a lock wait also.

The next line shows that MARK is waiting for a lock on DBSPACE 16, and that MIKE is holding that lock. The WAIT STAT column here shows that MIKE is also in a lock wait.

The third and fourth lines show that MIKE is waiting for CAROL and CAROL is waiting for STEVE. STEVE's agent is in a communication wait. STEVE is delaying LEIGH, MARK, MIKE, and CAROL.

The final line shows that MARY is in a communication wait, and that CAROL is waiting for her. (STEVE and MARY are both holding the lock for which CAROL is waiting.)

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show lock graph leigh
F4 004 LOCK          LOCK          WAIT LOCK  DBSP  LOCK    REQ    REQ
F4 004 REQUESTER  HOLDER      STAT TYPE  NUMBR  QUALFIER STATE  MODE  DUR
F4 004 2  LEIGH  4  MARK  LOCK  DBSP  15          G WAIT  S   LONG
F4 004 4  MARK  5  MIKE  LOCK  DBSP  16          G WAIT  X   LONG
F4 004 5  MIKE  6  CAROL LOCK  PAGE  21    88      G WAIT  IS  LONG
F4 004 6  CAROL 1  STEVE COMM  DBSP  1          G WAIT  IX  LONG
F4 004 6  CAROL 3  MARY  COMM  DBSP  1          G WAIT  IX  LONG
F4 004 ARI0065I Operator command processing is complete.

```

Figure 59. A Typical SHOW LOCK GRAPH Display on a VSE Operator Console

```

show lock graph leigh
LOCK          LOCK          WAIT LOCK  DBSP  LOCK    REQ    REQ
REQUESTER  HOLDER      STAT TYPE  NUMBR  QUALFIER STATE  MODE  DUR
2  LEIGH  4  MARK  LOCK  DBSP  15          G WAIT  S   LONG
4  MARK  5  MIKE  LOCK  DBSP  16          G WAIT  X   LONG
5  MIKE  6  CAROL LOCK  PAGE  21    88      G WAIT  IS  LONG
6  CAROL 1  STEVE COMM  DBSP  1          G WAIT  IX  LONG
6  CAROL 3  MARY  COMM  DBSP  1          G WAIT  IX  LONG
ARI0065I Operator command processing is complete.

```

Figure 60. A Typical SHOW LOCK GRAPH Display on a VM Operator Console or ISQL Terminal

SHOW LOCK MATRIX



The SHOW LOCK MATRIX command displays information about the usage of lock request blocks, and displays a lock wait table. It is optional to type the keyword MATRIX. You can enter SHOW LOCK and get the same results.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different DB2 Server for VSE & VM authorization IDs, within separate CMS units of work with one work unit waiting for another. To see if a user is in this wait state, enter the SHOW USERS command. See Figure 83 on page 137 for information on what to do when a VM user is in a wait state for a resource or database under one or more authorization IDs.

Lock request blocks are the internal control blocks that the database manager uses for data sharing. The following information is displayed about lock request block usage:

NLRBS

is the total number of lock request blocks that were defined. You specify this value for the NLRBS parameter when you start the application server.

IN USE

is the number of lock request blocks that are currently in use by the database manager. When IN USE starts to approach NLRBS, the database manager starts a process called escalation. (The process starts when IN

USE reaches NLRBS - 2*NCUSERS.) The escalation process attempts to free lock request blocks by exchanging lower-level locks (such as page or table locks) for a higher-level lock (a dbspace lock). Escalation can cause reduced sharing of data and rollback of some applications because of deadlock.

FREE is the number of lock request blocks that are currently free or available for use by the system for applications.

NLRBU

is the user limit of lock request blocks. You specify this value in the NLRBU parameter when you start the database manager. When an application reaches this limit, the application server starts the escalation process.

MAX USED BY LUW

is the maximum number of lock request blocks used by any one application. For example, if three applications accessed the database manager and required 200, 285, and 100 lock request blocks respectively, 285 would be displayed for MAX USED BY LUW.

If you are monitoring IN USE and MAX USED BY LUW, and only one application is active, MAX USED BY LUW is generally greater than IN USE. An application does not keep all of the lock request blocks while it runs. An application will keep some lock request blocks until it issues an SQL COMMIT WORK or ROLLBACK WORK but frees others after a very short time.

The application server resets MAX USED BY LUW to 0 any time an application reaches either the NLRBU or NLRBS limit and the lock escalation process occurs.

You can use MAX USED BY LUW to determine the lock request block requirements for preprocessing a DB2 Server for VSE & VM application program or for running an application that uses many locks. You use this support when the application server is running with NCUSERS=1.

The lock wait table is an n x n matrix where n is the number of users (NCUSERS). This matrix shows which users are waiting for locks and which are holding them. You can use this display to find users who are creating a bottleneck. Note that both SHOW LOCK GRAPH and SHOW LOCK MATRIX show the lock hierarchy, but they show it in different representations.

When the checkpoint agent is in lock contention because of a log archive checkpoint, the checkpoint agent is shown in the matrix along with the user agents. The database manager displays the agent identifier and authorization ID for the checkpoint agent in the first row and column of the matrix. C is the agent identifier, and CHECKPT is the authorization ID.

The entry in the matrix is the dbspace number where the lock contention has occurred. You can use this number on the SHOW LOCK DBSPACE command to find out more about the locking occurring in that dbspace. If lock contention has occurred on an internal system lock, the entry in the matrix is SYS or DB. Refer to "SHOW LOCK ACTIVE" on page 93 for more detail on these lock types. If the entry in the matrix shows lock contention, but no agent authorization ID, the application server is in the process of verifying CONNECT authority for the authorization ID.

In the console examples that follow, user agent 4 (MARY) is waiting for a lock that user agent 1 (CAROL) is holding. The lock is for some resource in dbspace number

12.

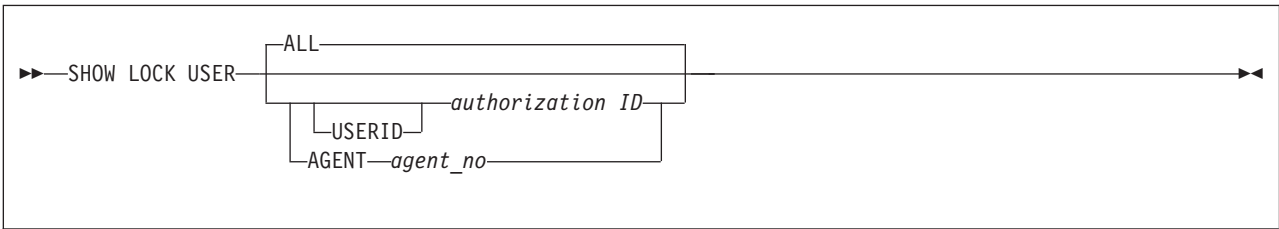
```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show lock
F4 004 Lock Request Block (LRB) and Lock Status:
F4 004  NLRBS    IN USE    FREE    NLRBU    MAX USED BY LUW
F4 004  -----
F4 004  2510     490      2020    1000     208
F4 004      *** THE LOCKWAIT TABLE ***
F4 004  ENTRY = DBSPACE NUMBER ON WHICH THERE IS LOCK CONTENTION
F4 004  The presence of an entry shows
F4 004  the agent requesting the lock and
F4 004  the agent contending for or holding the lock.
F4 004  AGENT          AGENT CONTENDING FOR OR HOLDING THE LOCK
F4 004  REQUESTING
F4 004  LOCK
F4 004          1      2      3      4      5
F4 004          CAROL  BRUCE  EDWARD  MARY
F4 004 1  CAROL  .....
F4 004 2  BRUCE  .....
F4 004 3  EDWARD .....
F4 004 4  MARY   12  .....
F4 004 5  .....
F4 004
F4 004 ARI0065I Operator command processing is complete.
```

Figure 61. A Typical SHOW LOCK MATRIX Display on a VSE Operator Console

```
show lock
Lock Request Block (LRB) and Lock Status:
 NLRBS    IN USE    FREE    NLRBU    MAX USED BY LUW
-----
 2510     490      2020    1000     208
      *** THE LOCKWAIT TABLE ***
ENTRY = DBSPACE NUMBER ON WHICH THERE IS LOCK CONTENTION
The presence of an entry shows
the agent requesting the lock and
the agent contending for or holding the lock.
AGENT          AGENT CONTENDING FOR OR HOLDING THE LOCK
REQUESTING
LOCK
          1      2      3      4      5
          CAROL  BRUCE  EDWARD  MARY
1  CAROL  .....
2  BRUCE  .....
3  EDWARD .....
4  MARY   12  .....
5  .....
ARI0065I Operator command processing is complete.
```

Figure 62. A Typical SHOW LOCK MATRIX Display on a VM Operator Console or ISQL Terminal

SHOW LOCK USER



The `SHOW LOCK USER` command displays locking information for a set of users. You can get locking information for all users that are active within the database manager by specifying the `ALL` parameter. `ALL` is the default. For information about a specific user, specify either the authorization ID or the agent number for the user. If you specify an agent number, you must specify the `AGENT` keyword. The `USERID` keyword is optional when you are specifying an authorization ID.

You can obtain information about users that are waiting with the `SHOW LOCK WANTLOCK` and `SHOW LOCK MATRIX` commands.

Note: If you enter a nonunique authorization ID, the `SHOW LOCK USER` command displays information about the lowest numbered agent that is processing for the specified authorization ID. This may not be the agent you are interested in. If it is not, enter a `SHOW ACTIVE` command to obtain the agent number. Then reenter the `SHOW LOCK USER` command and specify the agent number.

To get information about the checkpoint agent, specify either the letter `C` for the agent number or `CHECKPT` for the authorization ID.

There is one line of output for each resource or database that is locked by or is pending locking by a particular user. The lines are sorted by dbspace number and, when you specify `ALL`, by agent number. The following is displayed for each user:

AGENT

is the agent number or identifier.

The agent identifier for the checkpoint agent is `C`.

USER is the authorization ID of the person who is currently using the agent. In VM, the authorization ID is typically the user's virtual machine user ID.

Note that the authorization ID of the checkpoint agent is `CHECKPT`.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different authorization IDs, within separate CMS units of work, with one work unit waiting for another. To see if a user is in this wait state, enter the `SHOW USERS` command. See Figure 83 on page 137 for information on what to do when a VM user is in a wait state for a resource or database under one or more authorization IDs.

DBSPACE NUMBER

is the number of the dbspace for which information is being displayed. If the `LOCK TYPE` is `DB` or `SYS`, blanks appear in this column. The number corresponds to the `DBSPACENO` column of `SYSTEM.SYSCATALOG` and `SYSTEM.SYSDBSPACES`. You see information for each dbspace where the user has acquired or requested locks.

Note: For internal dbspaces (lock type is INT) a unique dspace identifier is displayed. This identifier is generated internally and does not correspond to the DBSPACENO column of SYSTEM.SYSCATALOG and SYSTEM.SYSDBSPACES.

LOCK TYPE

is the type of database resource that is locked by or is pending locking by the user. The LOCK TYPE can be one of the following:

DB for a lock on the database.

DBSP for a lock on an entire dspace.

IKEY for a lock on a key of an index.

INT for a lock on an entire internal dspace. There is never lock contention for internal dbspaces.

IPAG for a lock on a page of an index.

PAGE for a lock on a page (4 096 bytes) of data.

ROW for a lock on a row of data.

SYS for a special internal system lock.

TABL for a lock on an entire table.

Note: A database lock is acquired with long duration in intention exclusive (IX) mode at the start of every logical unit of work. A special internal system lock can actually be a latch on a page or block buffer. This will show as a LATCH WAIT if the SHOW ACTIVE command is entered.

The database lock is for log archive checkpoints. The database manager always takes a checkpoint before creating a log archive or switching to an alternate log. The checkpoint agent acquires a database lock with short duration in exclusive (X) mode when doing the checkpoint. The checkpoint agent's acquisition of the database lock ensures that no logical units of work are active when the checkpoint that precedes the log archive is in progress. The checkpoint agent releases the database lock at the end of the checkpoint.

(One line is displayed for each active lock type within a dspace.)

IN is the number of locks of the indicated type that are intention none locks.

SIX is the number of locks of the indicated type that are share and intention exclusive locks.

IS is the number of locks of the indicated type that are intention share locks.

IX is the number of locks of the indicated type that are intention exclusive locks.

S is the number of locks of the indicated type that are share locks.

U is the number of locks of the indicated type that are update locks.

X is the number of locks of the indicated type that are exclusive locks.

Z is the number of locks of the indicated type that are super exclusive locks.

For more information about lock modes, see the *DB2 Server for VSE & VM Diagnosis Guide and Reference*.

NUMBER WAITERS

is the number of users waiting for a lock of the type indicated. One of these waiters may be the user for whom the SHOW LOCK USER command was entered; that is, the lock has not yet been granted.

The following examples show the output of the SHOW LOCK USER command.

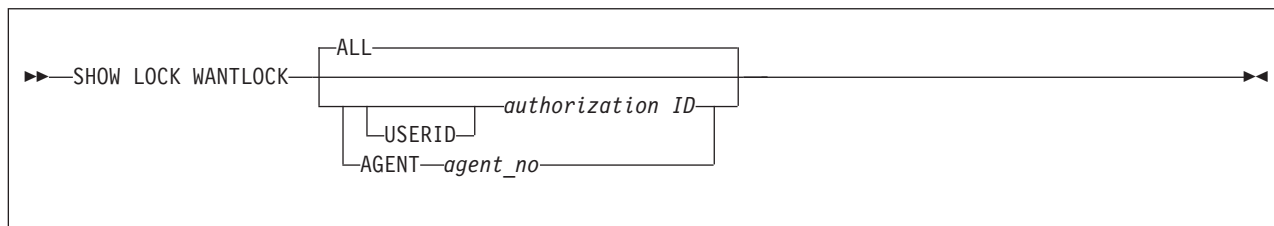
```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show lock user jessica
F4 004
F4 004 AGENT USER      DBSPACE LOCK
F4 004 7      JESSICA      DBSP 1 0 0 2 0 0 0 0 0 0 0
F4 004 7      JESSICA      IKEY 0 0 0 0 0 0 0 2 0 0 0
F4 004 7      JESSICA      ROW 0 0 0 0 0 0 0 1 0 0 0
F4 004 7      JESSICA      TABL 1 0 0 1 0 0 0 0 0 0 1
F4 004 ARI0065I Operator command processing is complete.
```

Figure 63. A Typical SHOW LOCK USER Display on a VSE Operator Console

```
show lock user all
AGENT USER      DBSPACE LOCK
NUMBER TYPE IN SIX IS IX S U X Z NUMBER
WAITERS
1 SQLDBA      DB 0 0 0 1 0 0 0 0 0 0 0
1 SQLDBA      1 DBSP 0 0 1 0 0 0 0 0 0 0 0
1 SQLDBA      1 ROW 0 0 0 0 1 0 0 0 0 0 0
1 SQLDBA      1 TABL 0 0 1 0 0 0 0 0 0 0 0
1 SQLDBA      7 DBSP 0 0 0 1 0 0 0 0 0 0 0
1 SQLDBA      7 IKEY 0 0 0 0 0 0 0 2 0 0 0
1 SQLDBA      7 ROW 0 0 0 0 0 0 0 1 0 0 0
1 SQLDBA      7 TABL 0 0 0 1 0 0 0 0 0 0 0
2 SQLUSRH      DB 0 0 0 1 0 0 0 0 0 0 0
2 SQLUSRH      1 DBSP 0 0 1 0 0 0 0 0 0 0 0
2 SQLUSRH      1 ROW 0 0 0 0 1 0 0 0 0 0 0
2 SQLUSRH      1 TABL 0 0 1 0 0 0 0 0 0 0 0
2 SQLUSRH      7 DBSP 1 0 0 0 0 0 0 0 0 0 0
2 SQLUSRH      7 TABL 1 0 0 0 0 0 0 0 0 0 1
3 SQLUSRJH      DB 0 0 0 1 0 0 0 0 0 0 0
3 SQLUSRJH      1 DBSP 0 0 1 0 0 0 0 0 0 0 0
3 SQLUSRJH      1 ROW 0 0 0 0 1 0 0 0 0 0 0
3 SQLUSRJH      1 TABL 0 0 1 0 0 0 0 0 0 0 0
3 SQLUSRJH      7 DBSP 0 0 0 1 0 0 0 0 0 0 0
ARI0065I Operator command processing is complete.
```

Figure 64. A Typical SHOW LOCK USER Display on a VM Operator Console or ISQL Terminal

SHOW LOCK WANTLOCK



The SHOW LOCK WANTLOCK command displays information about a lock that an agent wants. The command also displays the queue of requests for the lock, including the user who is currently holding the lock.

To display information about all locks currently wanted by all active agents, specify ALL (the default). When you specify ALL, one line is displayed for each user that is in a lock wait.

To display information about the lock that a particular agent wants, specify either the agent number or the authorization ID for which the agent is doing work.

Note: If you enter a nonunique authorization ID, the SHOW LOCK WANTLOCK command displays information about the lowest numbered agent that is processing for the specified authorization ID. This may not be the agent you are interested in. If not, enter a SHOW ACTIVE command to obtain the agent number. Then reenter the SHOW LOCK WANTLOCK command with the agent number specified.

If you specify an agent number, you must specify the AGENT keyword. The USERID keyword is optional when you are specifying an authorization ID. When an authorization ID or agent number is specified, the application server displays a waiting list of all users holding the lock and all users waiting for the lock. They are shown in the order in which they are waiting.

To display information about the checkpoint agent, specify either the letter C for the agent number or CHECKPT for the authorization ID.

The following is displayed for each lock that the agent (or agents) wants:

AGENT

is the agent number (identifier).

Note that the agent identifier for the checkpoint agent is the letter C.

USER is the authorization ID of the person who is currently using the agent. In VM, the authorization ID is typically the user's virtual machine authorization ID.

Note that the authorization ID of the checkpoint agent is CHECKPT.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different authorization IDs, within separate CMS work units, with one work unit waiting for another. To see if a user is in this wait state, enter the SHOW USERS command. See Figure 83 on page 137 for information on what to do when a VM user is in a wait state for a resource or database under one or more authorization IDs.

LOCK TYPE

is the type of database resource that the user wants to lock. LOCK TYPE can be one of the following:

DB for a lock on the database.

DBSP for a lock on an entire dbspace.

IKEY for a lock on a key of an index.

IPAG for a lock on a page of an index.

PAGE for a lock on a page (4 096 bytes) of data.

ROW for a lock on a row of data.

SYS for a special internal system lock.

TABL for a lock on an entire table.

Note: A database lock is acquired with long duration in intention exclusive (IX) mode at the start of every logical unit of work. A special internal system lock can actually be a latch on a page or block buffer. This will show as a LATCH WAIT if the SHOW ACTIVE command is entered.

The database lock is for log archive checkpoints. The application server always takes a checkpoint before creating a log archive or switching to an alternate log. The checkpoint agent acquires a database lock with short duration in exclusive (X) mode when doing the checkpoint. The checkpoint agent's acquisition of the database lock ensures that no logical units of work are active when the checkpoint that precedes the log archive is in progress. The checkpoint agent releases the database lock at the end of the checkpoint.

DBSPACE NUMBER

is the number of the dbspace where the agent wants to acquire a lock. If the LOCK TYPE is DB or SYS, blanks appear in this column. The number corresponds to the DBSPACENO column of SYSTEM.SYSCATALOG and SYSTEM.SYSDBSPACES.

LOCK QUALIFIER

is an entity that the database uses internally to qualify the lock that is wanted. The LOCK QUALIFIER varies depending on LOCK TYPE:

LOCK TYPE LOCK QUALIFIER

DB blanks.

DBSP blanks, because the dbspace number (which is the qualifier) is already displayed under DBSP NUMBR.

PAGE the page number that the database manager has previously assigned to the page.

TABL the internal identifier that the database manager uses for the table. The values in TABL correspond to those in the TABID column of SYSTEM.SYSCATALOG.

ROW the internal identifier that the database manager uses for the row.

IKEY an internal value that the database manager uses to represent the index key. The Lock Qualifier can be an internal ROW identifier and an internal INDEX identifier if the index is unique and KEY LEVEL Locking is in effect.

IPAG the index page number. As with data page numbers, the application server assigns numbers to pages that contain indexes.

SYS blanks.

The values PAGE, ROW, IKEY, and IPAG are useful only for pattern matching during problem determination. The DBSP and TABL values, however, correspond to the DBSPACENO and TABID columns in the catalogs. For example, the following SELECT statement identifies the table

by using the DBSP and TABL values:

```
SELECT TABID, TNAME FROM SYSTEM.SYSCATALOG
WHERE TABID=tbl_value AND DBSPACENO=dbsp_value
```

REQUEST STATE

the status of the requested lock. REQUEST STATE can be:

GRANT

means that the agent has requested the lock and has received it. The agent is currently holding the lock (that is, not waiting for it).

C WAIT

means waiting to be converted. The agent has made a lock request, but the lock cannot be allocated now. The agent has already locked a database resource in a particular lock mode. Now the agent wants to change the mode of the lock it acquired, so it issues a new lock request. This could happen, for example, when a user is doing read-only work during a logical unit of work, and then wants to modify the data. The application server must change a read-only lock to a write lock. This is called lock conversion.

If another user also held a lock on the resource, the requested mode could be incompatible. A conversion is not immediately possible. The user waits because the lock is waiting to be converted.

G WAIT

means waiting to be granted. The agent made a lock request, but someone else has already locked the resource. The agent is waiting to be granted the lock.

DENIED

means a requested lock was not available, and the database manager canceled the request. That is, the lock was denied and the agent is not waiting for it.

In this case, the agent requested a lock that was not available. The agent indicated by its request that, if no lock was available, it would not wait for the lock. The database manager denied the request rather than put the agent in a lock wait. An example of this kind of request is a user trying to access a private dbspace that is currently in use. If the private dbspace is currently in use and the lock request is not compatible, the database manager does not let the lock request wait. Instead the application server denies the request.

REQUEST MODE

is the mode of the lock for which the user is waiting:

- IN** indicates that the user requested an intention none lock.
- SIX** indicates that the user requested a share and intention exclusive lock.
- IS** indicates that the user requested an intention share lock.
- IX** indicates that the user requested an intention exclusive lock.
- S** indicates that the user requested a share lock.

- U** indicates that the user requested an update lock.
- X** indicates that the user requested an exclusive lock.
- Z** indicates that the user requested a super exclusive lock.

For more information about lock modes, see the *DB2 Server for VSE & VM Diagnosis Guide and Reference*.

DUR (Duration)

indicates the duration of the lock that the waiter requested. DUR can be:

- INST (instant)** used to test the request state of the lock. That is, it is used to test if a particular database resource (an index key, for example) is in use. The database manager frees the lock when the request is granted.
- LONG** indicates that when the lock is granted, it will be held until the end of the logical unit of work.
- MED (medium)** indicates that when the lock is granted, it will be held until the database manager explicitly releases it, or until the end of the logical unit of work.
- SHORT** indicates that the waiter will not require this lock until the logical unit of work ends. Instead, the lock will be released at the end of the call to the database storage subsystem (DBSS).

The following examples show the output from the SHOW LOCK WANTLOCK command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show lock wantlock
F4 004
F4 004 AGENT  USER      LOCK  DBSPACE  LOCK      REQUEST  REQUEST
F4 004 5    CLIFF    PAGE  12      88        G WAIT   S        LONG
F4 004 6    DOT     PAGE  12      88        G WAIT   X        LONG
F4 004 4    MIKE    PAGE  12      88        G WAIT   S        LONG
F4 004 2    BUTCH   PAGE  12      88        G WAIT   X        LONG
F4 004 ARI0065I Operator command processing is complete.

```

Figure 65. A Typical SHOW LOCK WANTLOCK Display on a VSE Operator Console

```

show lock wantlock
AGENT  USER      LOCK  DBSPACE  LOCK      REQUEST  REQUEST  DURATION
5     CLIFF    PAGE  12      88        G WAIT   S        LONG
6     DOT     PAGE  12      88        G WAIT   X        LONG
4     MIKE    PAGE  12      88        G WAIT   S        LONG
2     BUTCH   PAGE  12      88        G WAIT   X        LONG
ARI0065I Operator command processing is complete.

```

Figure 66. A Typical SHOW LOCK WANTLOCK Display on a VM Operator Console or ISQL Terminal

SHOW LOG

▶—SHOW LOG—◀

The SHOW LOG command displays global log control information and status as follows:

- Active log (if alternate logging is enabled).
- Inactive log (if alternate logging is enabled) and whether it has been archived or is empty.
- Log size. The total size of the log in bytes.
- Log usage. The percentage of log space used and the number of bytes used.
- If archiving is disabled (that is, LOGMODE=Y or N), SHOW LOG displays the number of bytes and the percentage of the log remaining before SLOGCUSH (the DB2 Server for VSE & VM log cushion) is reached. It also displays a list of agents that were active before the last checkpoint.

SLOGCUSH is a DB2 Server for VSE & VM initialization parameter value. When the log reaches this cushion, the log overflow procedure begins. This procedure rolls back the oldest logical unit of work that began before the last checkpoint.

- If archiving is enabled (LOGMODE=A or L), and an archive is not taking place, you see the number of bytes and the percentage of the log remaining before ARCHPCT is reached. If alternate logging is not enabled and ARCHPCT is reached, the application server begins an archive. If alternate logging is enabled and ARCHPCT is reached, the application server attempts to switch to the inactive log if possible. (ARCHPCT is a DB2 Server for VSE & VM initialization parameter.)
- The number of log pages is displayed if there are any log pages remaining before a checkpoint. This is not displayed if an archive is taking place. A message is displayed if a checkpoint is pending. The DB2 Server for VSE & VM initialization parameter CHKINTVL determines how many log pages are written between checkpoints.

The following examples show the output from the SHOW LOG command.

```
AR
MSG F4
AR 015 1I40I READY
F4 004 ARI0062A SQLDS :
F4-004          Enter a DB2 Server for VSE operator command.
4 show log
F4 004 Log Status:
F4 004 Active Log: ALTLGD1
F4 004 Inactive Log: LOGDSK1 - is not archived
F4 004   Log size is 700416 bytes.
F4 004   Log used is 3 percent 23326 bytes.
F4 004   Log remaining before overflow procedure is 86 percent
F4 004 607048 bytes.
F4 004   No non-R/O agents began before the last checkpoint.
F4 004   Archiving is not enabled.
F4 004   Log pages remaining before checkpoint is 5.
F4 004 ARI0065I Operator command processing is complete.
```

Figure 67. A Typical SHOW LOG Display on a VSE Operator Console

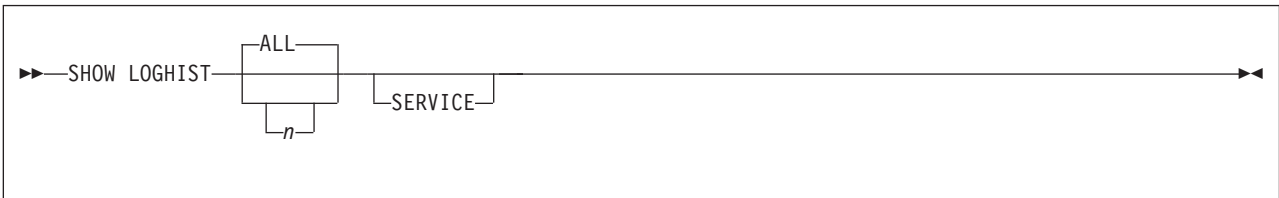
```

show log
Log Status:
F4 004 Active Log: ALTLGD1
F4 004 Inactive Log: LOGDSK1 - is not archived
  Log size is 700416 bytes.
  Log used is 3 percent 23326 bytes.
  Log remaining before overflow procedure is 86 percent
607048 bytes.
  No non-R/O agents began before the last checkpoint.
  Archiving is not enabled.
  Log pages remaining before checkpoint is 5.
ARI0065I Operator command processing is complete.

```

Figure 68. A Typical SHOW LOG Display on a VM Operator Console or ISQL Terminal

SHOW LOGHIST



The log history is a 4-kilobyte page at the end of the log file. Its layout consists of a header area followed by multiple entries, each recording a particular log-related event. These events usually fall into one of the following categories:

- Database or log archives and restores
- Changes to the LOGMODE startup parameter value
- Log initialization using SQLLOG (COLDLOG).

For more information on the history area, see the *DB2 Server for VSE System Administration*, or *DB2 Server for VM System Administration*.

The SHOW LOGHIST command lets you view the log history without having to manually record the information.

The following rules apply to the parameters of the SHOW LOGHIST command:

- If you omit the first parameter, ALL is assumed. The ALL parameter displays all used entries of the history area.
- If you enter a number (*n*) as the first parameter, the SHOW LOGHIST command displays only the most recent *n* entries. If *n* is greater than the number of entries, *n* is equivalent to ALL.
- If you specify the SERVICE parameter, the SHOW LOGHIST command displays information generally of interest only to IBM service personnel (the FLAGS heading). All other output appears whether you specify the SERVICE parameter or not.

The first line of output for the SHOW LOGHIST command has three possible responses:

Created last database archive on mm-dd-yyyy at hh:mm:ss at entry n
 gives the date and time of the most recent database manager or user database archive, if any are recorded in the history area. You can use this information to find the corresponding *sys db arch created* or *user db arch created* entry in the list. You can then use this information to determine the events since that archive.

The most recent archive date and time will be displayed even if you request too few entries to have the output extend as far back as the time of that archive.

The *entry number* (1 represents the first entry listed) is shown to allow you to resubmit the command, if necessary, to display as far back as that archive.

Restored database archive at entry n. Database archive was created on mm-dd-yyyy at hh:mm:ss

gives the date and time of the most recent database restore if one has been performed since the last database archive. You can use this information to find the corresponding restore set entries. You will require this information if the database archive requires restoring again.

No database archives recorded

indicates that no database archives are recorded in the history area. This result can occur in one of three ways:

- The database has not been archived since the database was generated.
- The database has been archived at least once and you have since done a log reconfiguration. A log reconfiguration erases the history area.
- The history area became full and the most recent database archive entry was overwritten. Entries are not often overwritten. You are warned when the history area becomes full to the ARCHPCT value specified during initialization.

ENTRY TYPE

is the type of entry shown on that line. It indicates what activity is recorded in the entry. The user specifies the number of entries shown. These entries can be the following:

Coldlog reformat

indicates a coldlog was done to reformat the logs. In this case, the history area remains intact.

Coldlog reconfig

indicates a coldlog was done to reconfigure the logs. In this case, the history area was erased.

Logmode N indicates that the logmode was changed to N.

Logmode Y indicates that the logmode was changed to Y.

Logmode A indicates that the logmode was changed to A.

Logmode L indicates that the logmode was changed to L.

Sys db arch created

indicates that a database archive was created.

User db arch created

indicates a user archive was created. The date and time indicate when the SQLEND UARCHIVE command shut the system down, not when the archive was actually performed.

Log swtch - &a to &b

indicates that the active log has changed from &a to &b. &a and &b can be pri (for LOGDSK1) or alt (for ALTLGD1).

End of restore set

indicates the end of a restore set. The continuity of log archives is broken. A database restore, either database manager or user, follows this entry.

Sys db arch restored

indicates the database archive, identified by its date and time, was successfully restored.

User db arch restored

indicates the user archive, identified by its date and time, was successfully restored.

Log arch created

indicates a log archive was created.

Log arch crt by restore

indicates a log archive was created to save the active log because a database restore, either database manager or user, was initiated. This log will be the last member of the restore set for the last database archive. If the last database archive was restored, this log archive would not be listed among the restored logs because its contents were restored to the active log before startup.

Log arch restored

indicates that the log archive, identified by its date and time, was successfully restored.

DATE is the date of the entry in system date format. There are two exceptions:

- In an entry for a database restore, DATE is the date on which the restored archive was created.
- Before Version 3 Release 1 of the database manager, the date displayed for the following entries was 00-00-0000:
Coldlog
Logmode changes
End of restore set.

TIME is the time at which the entry was made, in 24-hour clock format. There are two exceptions:

- In an entry for a restore, TIME is the time at which the restored archive was created.
- Prior to Version 3 Release 1 of the database manager, the time displayed for the following entries was 00-00-00:
Coldlog
Logmode changes
End of restore set.

Note: Before Version 3 Release 1, DATE and TIME entries for ARCHIVE and RESTORE, with the exception of End of restore set, displayed date and time information respectively.

FLAGS

shows the value of some internal-use flags intended for IBM service use only. The heading and data are blank unless you specify the SERVICE parameter.

LOG FN (Valid for VM only)

is the file name of the output disk file for a log archived to disk.

LOG FT (Valid for VM only)

is the file type of the output disk file for a log archived to disk.

The following examples show the output from the SHOW LOGHIST command.

```
AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show loghist all service
F4 004
F4 004 Restored database archive at entry 2.
F4 004 Database archive was created on 12-29-1996 at 14:42:43.
F4 004
F4 004 Entry Type           Date           Time           Flags
F4 004 Log arch restored    12-29-1996   14:54:59   B000
F4 004 User db arch restored 12-29-1996   14:42:43   A000
F4 004 End of restore set    12-30-1996   13:21:57   0000
F4 004 Log arch created      12-29-1996   14:54:59   9000
F4 004 User db arch created  12-29-1996   14:42:43   8000
F4 004 Log arch restored    12-10-1996   16:19:09   B800
F4 004 Log arch restored    12-10-1996   16:18:32   B000
F4 004 Sys db arch restored  12-10-1996   16:11:25   A000
F4 004 End of restore set    12-10-1996   16:20:49   0000
F4 004 Log arch crt by restore 12-10-1996   16:20:49   9000
F4 004 Log arch created      12-10-1996   16:19:09   9000
F4 004 Log swtch - alt to pri 12-10-1996   16:19:00   9000
F4 004 Log arch created      12-10-1996   16:18:32   9000
F4 004 Log swtch - pri to alt 12-10-1996   16:17:32   9000
F4 004 Logmode L             12-10-1996   16:16:44   0000
F4 004 Sys db arch created   12-10-1996   16:11:25   8000
F4 004 Logmode Y             12-01-1996   08:45:33   0000
F4 004 Logmode N             11-31-1996   09:15:08   0000
F4 004 Coldlog reformat     11-29-1996   13:35:51   0000
F4 004 ARI0065I Operator command processing is complete.
```

Figure 69. A Typical SHOW LOGHIST Display on a VSE Operator Console

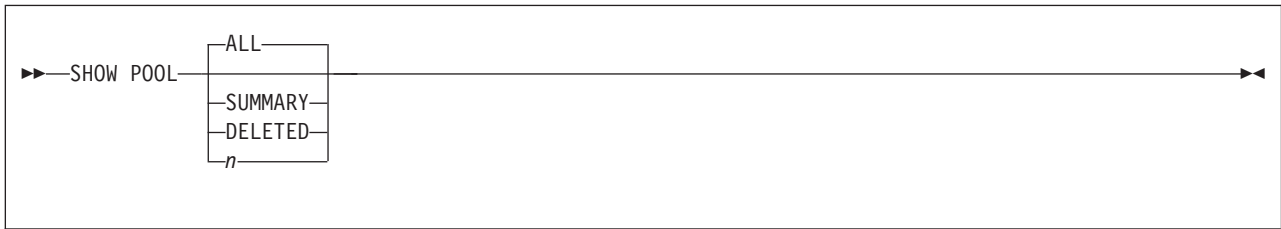
```
show loghist all service

Restored database archive at entry 2.
Database archive was created on 12-29-1996 at 14:42:43.

Entry Type           Date           Time           Flags   Log FN   Log FT
Log arch restored    12-29-1996   14:54:59   B000   SQLMACHO 12298901
User db arch restored 12-29-1996   14:42:43   A000
End of restore set    12-30-1996   13:21:57   0000
Log arch created      12-29-1996   14:54:59   9000   SQLMACHO 12298901
User db arch created  12-29-1996   14:42:43   8000
Log arch restored    12-10-1996   16:19:09   B800   SQLMACHO 12108902
Log arch restored    12-10-1996   16:18:32   B000   SQLMACHO 12108901
Sys db arch restored  12-10-1996   16:11:25   A000
End of restore set    12-10-1996   16:20:49   0000
Log arch crt by restore 12-10-1996   16:20:49   9000   SQLMACHO 12108903
Log arch created      12-10-1996   16:19:09   9000   SQLMACHO 12108902
Log swtch - alt to pri 12-10-1996   16:19:00   9000   SQLMACHO 12108902
Log arch created      12-10-1996   16:18:32   9000   SQLMACHO 12108901
Log swtch - pri to alt 12-10-1996   16:17:32   9000   SQLMACHO 12108901
Logmode L             12-10-1996   16:16:44   0000
Sys db arch created   12-10-1996   16:11:25   8000
Logmode Y             12-01-1996   08:45:33   0000
Logmode N             11-31-1996   09:15:08   0000
Coldlog reformat     11-29-1996   13:35:51   0000
ARI0065I Operator command processing is complete.
```

Figure 70. A Typical SHOW LOGHIST Display on a VM Operator Console or ISQL Terminal

SHOW POOL



A storage pool is made up of one or more database extents. A dbextent is the physical medium in which the application server stores data. Internally, all database extents look like a continuous DASD address space of physical 4096-byte (4-kilobyte) pages.

The SHOW POOL command displays physical storage information about each storage pool defined. You can use this command to monitor physical storage usage.

The following examples show the output of the SHOW POOL command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show pool
F4 004
F4 004 POOL NO.  1:          NUMBER OF EXTENTS =  3
F4 004
F4 004 EXTENT  TOTAL    NO. OF      NO. OF      NO. OF      %
F4 004 NO.     PAGES   PAGES USED  FREE PAGES  RESV PAGES  USED
F4 004  1      285    274         11          0           96
F4 004  2      285     33         252         0           11
F4 004  6      741     0          741         0            0
F4 004 TOTAL  1311    307        1004        20          23
F4 004
F4 004 POOL NO.  2:          NUMBER OF EXTENTS =  3
F4 004
F4 004 EXTENT  TOTAL    NO. OF      NO. OF      NO. OF      %
F4 004 NO.     PAGES   PAGES USED  FREE PAGES  RESV PAGES  USED
F4 004  3      114     2           112         0            1
F4 004  4      114     0           114         0            0
F4 004  9      114     0           114         0            0
F4 004 TOTAL  342     2           340        20           0
F4 004
F4 004 POOL NO.  4:          NUMBER OF EXTENTS =  1      SHORT ON STORAGE
F4 004
F4 004 EXTENT  TOTAL    NO. OF      NO. OF      NO. OF      %
F4 004 NO.     PAGES   PAGES USED  FREE PAGES  RESV PAGES  USED
F4 004 10      285    260         25          0           91
F4 004 TOTAL  285    260         25          0           91
F4 004
F4 004 FREE AREAS: NUMBER OF DELETED EXTENTS =  3
F4 004
F4 004 EXTENT  TOTAL
F4 004 NO.     PAGES
F4 004  (2+)    57
F4 004  5      171
F4 004  7      228
F4 004  8      342
F4 004  END    9552
F4 004 TOTAL  10350
F4 004
F4 004 Maximum number of DBEXTENTS = 64
F4 004 ARI0065I Operator command processing is complete.

```

Figure 71. A Typical SHOW POOL Display on a VSE Operator Console

```

show pool

POOL NO. 1:          NUMBER OF EXTENTS = 3

EXTENT  TOTAL  NO. OF  NO. OF  NO. OF  %
NO.     PAGES  PAGES USED  FREE PAGES  RESV PAGES  USED
1       285   274      11        0           96
2       285   33       252       0           11
6       741   0        741       0           0
TOTAL  1311   307     1004     20          23

POOL NO. 2:          NUMBER OF EXTENTS = 3

EXTENT  TOTAL  NO. OF  NO. OF  NO. OF  %
NO.     PAGES  PAGES USED  FREE PAGES  RESV PAGES  USED
3       114   2        112       0           1
4       114   0        114       0           0
9       114   0        114       0           0
TOTAL  342   2        340     20          0

POOL NO. 4:          NUMBER OF EXTENTS = 1          SHORT ON STORAGE

EXTENT  TOTAL  NO. OF  NO. OF  NO. OF  %
NO.     PAGES  PAGES USED  FREE PAGES  RESV PAGES  USED
10      285   260      25        0           91
TOTAL  285   260      25        20          91

FREE AREAS: NUMBER OF DELETED EXTENTS = 3

EXTENT  TOTAL
NO.     PAGES
(2+)   57
5      171
7      228
8      342
END    9552
TOTAL  10350

Maximum number of DBEXTENTS = 64
ARI0065I Operator command processing is complete.

```

Figure 72. A Typical SHOW POOL Display on a VM Operator Console or ISQL Terminal

The SHOW POOL command with the SUMMARY parameter gives output identical to the SHOW DBEXTENT command. The SHOW DBEXTENT command only displays information for pools.

The SHOW POOL command with the DELETED parameter displays all the free area on the directory's page map table, including deleted dbextents.

The SHOW POOL command with a pool number specified for n displays information for that pool.

When you enter the SHOW POOL command without specifying a parameter, the default is SHOW POOL ALL. The following examples show the default SHOW POOL command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show pool summary
F4 004 POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
F4 004 NO.      PAGES      PAGES USED  FREE PAGES  RESV PAGES  USED  EXTENTS  SOS
F4 004 1      1311      307      1004      20      23      3
F4 004 2      342      2      340      20      0      3
F4 004 4      285      260      25      20      91      1      *
F4 004 FREE  10350
F4 004 ARI0065I Operator command processing is complete.

```

Figure 73. A Typical SHOW POOL SUMMARY Display on a VSE Operator Console

```

show pool summary
POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
NO.      PAGES      PAGES USED  FREE PAGES  RESV PAGES  USED  EXTENTS  SOS
1      1311      307      1004      20      23      3
2      342      2      340      20      0      3
4      285      260      25      20      91      1      *
FREE  10350
ARI0065I Operator command processing is complete.

```

Figure 74. A Typical SHOW POOL SUMMARY Display on a VM Operator Console or ISQL Terminal

The SHOW POOL output values have the following meanings.

POOL NO.

is the number of the storage pool in which the dbextents reside. If the value of the number is positive, the storage pool is recoverable; that is, the storage pool is one for which the application server does recovery, through logging *every* change to the pool. (This is usually the situation.) The application server records *every* change to recoverable pools. If the value is negative, the storage pool is nonrecoverable. The application server does not log changes to a nonrecoverable pool. The application server does however, take a checkpoint after every LUW; therefore, *partial recovery* is still available. The absolute value of a negative storage pool number is the number of the storage pool. For example, -54 indicates that storage pool number 54 is nonrecoverable.

NUMBER OF EXTENTS

is the number of dbextents defined for that storage pool.

Note: Extra information is displayed beside this number because of the VMDSS code enabled. For more information see the *DB2 Server for VSE & VM Performance Tuning Handbook* manual.

SHORT ON STORAGE

is a flag indicating when the storage pool is running out of storage. An asterisk (*) appears in the SHORT ON STORAGE column of the SHOW POOL SUMMARY display when the storage pool is running out of storage (otherwise the column is blank). In the SHOW POOL ALL display, the words SHORT ON STORAGE appear to the right of POOL NO. when that storage pool is running out of storage. Otherwise, no words appear. The level that sets this flag is the value used for the SOSLEVEL parameter when you started the application server

Note: The SHORT ON STORAGE flag appears as SOS in the SHOW POOL SUMMARY display in Figure 73 on page 121.

TOTAL PAGES

is the total number of pages in the storage pool (or extent) assigned in the database directory.

Note: The application server calculates pages based on 57 pages per cylinder, rather than the actual number of pages per cylinder of the DASD.

NO. OF PAGES USED

is the number of pages within the storage pool (or extent) that have been acquired for use as data, header, or index pages.

NO. OF FREE PAGES

is the number of pages within the storage pool (or extent) that are not currently being used.

NO. OF RESV PAGES

is the number of reserved pages in the storage pool. A minimum of 20 pages is always reserved. See the *DB2 Server for VSE & VM Diagnosis Guide and Reference* for a description of reserved pages. Reserved pages apply only to the total number of pages in the storage pool.

% USED

is the percentage of pages in the storage pool (or extent) that are currently being used.

TOTAL

is the column total of all information presented in the various columns.

FREE AREAS

is a column of information about deleted dbextents. This information includes the following:

NUMBER OF DELETED DBEXTENTS

is the total number of dbextents that have been deleted and are available for reuse. When a deleted dbextent is reused it is no longer included in this total.

EXTENT NO.

is the number identifying the deleted dbextent. As an example, the value (2+) indicates that there is empty directory space between extents 2 and 3. Since there is no dbextent number associated with this space, to reclaim it you must delete dbextent 2 or dbextent 3.

END is the amount of contiguous unmapped area at the end of the database directory.

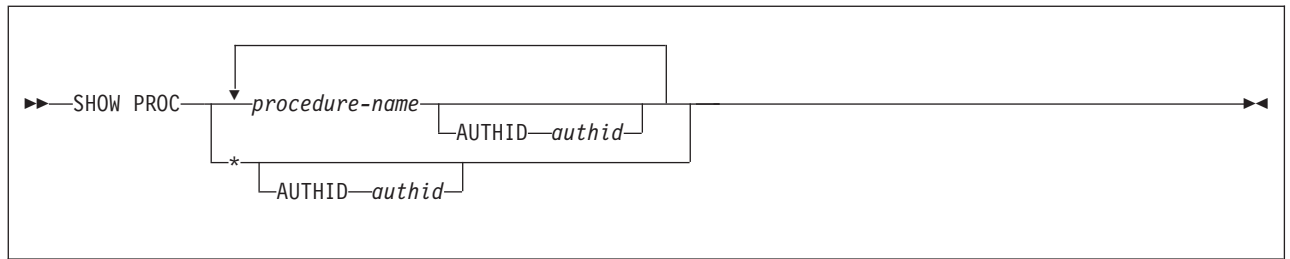
TOTAL PAGES

is the number of pages assigned in the database directory.

MAXIMUM NUMBER OF DBEXTENTS

is the largest number of individual dbextents you are allowed to have. This number is defined when the database is generated.

SHOW PROC



The `SHOW PROC` command displays cached information about stored procedures. Several output lines are displayed for each procedure. Note that if more than one version of a stored procedure exists and `AUTHID` is not specified, output is displayed for each version. For example, if there are two versions of `PROC1`, one for public use (that is, the `AUTHID` column in `SYSTEM.SYSROUTINES` is blank), and one for use by a specific individual (for example, the `AUTHID` column might contain `authid1`), output is displayed for both the public copy and the copy that is accessible only by `authid`. The following information is displayed:

- The name of the stored procedure
- The authorization ID allowed to run the procedure. If this field is blank, any user can run the procedure.
- The name of the load module or phase associated with the stored procedure. This field may contain blanks, if the stored procedure request is queued waiting for the `START PROC` command.
- The name of the stored procedure server group in which the procedure is running.
- The status of the stored procedure. The possible status values are:
 1. `STARTED`: where requests for the stored procedure can be processed
 2. `STOP-QUE`: where requests for the stored procedure are queued
 3. `STOP-REJ`: where requests for the stored procedure are rejected
- Whether the stored procedure is running in the default server group. Note that this information is displayed only if the stored procedure can run in the default server group, that is, if the column `DEFSERV` in `SYSTEM.SYSROUTINES` is `'Y'` or `NULL` for that procedure.
- The number of agents that are currently running the stored procedure
- The maximum number of agents that have ever been running the procedure concurrently, since the database manager was started
- The number of agents that are currently waiting to run the stored procedure
- The maximum number of agents that have ever been waiting to run the stored procedure, since the database manager was started
- The number of times an `SQL CALL` statement timed out while waiting for a server to be assigned for the stored procedure
- The number of times the stored procedure has terminated abnormally

procedure-name Identifies the procedure to be displayed. An asterisk (`'*`) contained in the last position of a procedure names acts as a wildcard. For example, `'ABC*` indicates that the command should be applied to all stored procedures whose names begin with `'ABC`.

authid The authorization ID for this stored procedure. This field is used for procedures that can be run only from one authorization ID. If no authorization ID is specified, information is displayed for all procedures with the name specified.

* Displays information for all stored procedures that are in the cache.

The following are examples of the SHOW PROC command.

```
SHOW PROC *
SHOW PROC PROC1
SHOW PROC PROC2 PROC20
SHOW PROC PROC2*
SHOW PROC MYPROC AUTHID USER1
```

The first statement above displays information for all stored procedures. The second example displays information for procedure PROC1 only. The third shows information for PROC2 and PROC20. The fourth statement displays information for any procedure that starts with 'PROC2'. The last statement displays information for the version of MYPROC that is accessible only by authorization ID USER1.

The following example shows the output from the SHOW PROC command.

```
show proc *
Status of stored procedure(s) 1997-09-30 08:56:42
Procedure: USERPRC1 AUTHID:
Module: MODULE1 Running in Group: PSRV1
Status: STARTED. Running in Default Group: N
Agents currently running this procedure: 1
Max num of agents that have run this proc concurrently: 1
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 1
Number of timeouts: 0
Number of abends: 0
Procedure: USERPRC2 AUTHID:
Module: MODULE2 Running in Group:
Status: STOP-QUE Running in Default Group: N
Agents currently running this procedure: 0
Max num of agents that have run this proc concurrently: 2
Agents waiting to run this procedure: 5
Max num of agents that have waited to run this proc: 5
Number of timeouts: 3
Number of abends: 1
Procedure: USERPRC3 AUTHID:
Module: MODULE3 Running in Group: PSRV1
Status: STARTED.
Agents currently running this procedure: 2
Max num of agents that have run this proc concurrently: 2
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 6
Number of timeouts: 0
Number of abends: 0
Procedure: USERPRC4 AUTHID:
Module: MODULE4 Running in Group:
Status: STOP-REJ Running in Default Group: N
Agents currently running this procedure: 0
Max num of agents that have run this proc concurrently: 1
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 1
Number of timeouts: 0
Number of abends: 0
ARI0065I Operator command processing is complete.
```

Figure 75. A Typical SHOW PROC Display on a VM Operator Console (Part 1 of 2)

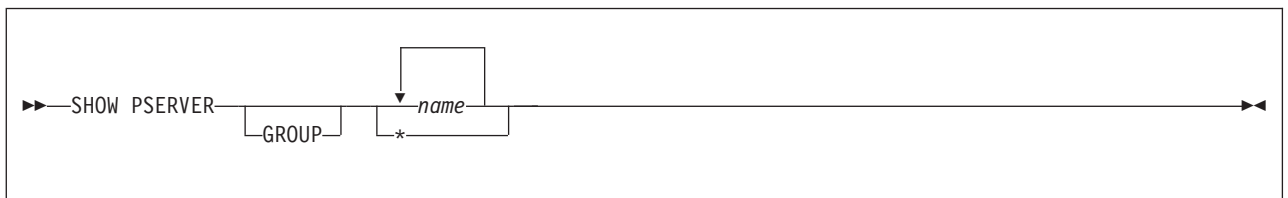
```

show proc *
Status of stored procedure(s) 1997-09-30 08:59:29
Procedure: PROC1 AUTHID: USERX
Module: PMOD1 Running in Group: PSRV2
Status: STARTED. Running in Default Group: N
Agents currently running this procedure: 1
Max num of agents that have run this proc concurrently: 1
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 1
Number of timeouts: 0
Number of abends: 0
Procedure: PROC2 AUTHID:
Module: PMOD2 Running in Group:
Status: STOP-REJ
Agents currently running this procedure: 0
Max num of agents that have run this proc concurrently: 2
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 2
Number of timeouts: 1
Number of abends: 1
Procedure: PROC3 AUTHID:
Module: PMOD3 Running in Group:
Status: STOP-QUE
Agents currently running this procedure: 0
Max num of agents that have run this proc concurrently: 1
Agents waiting to run this procedure: 3
Max num of agents that have waited to run this proc: 5
Number of timeouts: 0
Number of abends: 0
Procedure: TQUERY1 AUTHID:
Module: TMOD1 Running in Group: PSRV2
Status: STARTED. Running in Default Group: Y
Agents currently running this procedure: 2
Max num of agents that have run this proc concurrently: 4
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 2
Number of timeouts: 0
Number of abends: 0
Procedure: TQUERY2 AUTHID: USER2
Module: TMOD2 Running in Group: PSRV2
Status: STARTED.
Agents currently running this procedure: 1
Max num of agents that have run this proc concurrently: 1
Agents waiting to run this procedure: 0
Max num of agents that have waited to run this proc: 0
Number of timeouts: 0
Number of abends: 0
ARI0065I Operator command processing is complete.

```

Figure 75. A Typical SHOW PROC Display on a VM Operator Console (Part 2 of 2)

SHOW PSERVER



The SHOW PSERVER command displays cached information about stored procedure servers. The command displays one line for each server for which information is requested. Each line displays:

- The name of the stored procedure server

- The name of the group in which it resides
- The status of the server. The possible values:

STARTING

The START PSERVER command is in progress.

STARTED

The stored procedure server is running.

STOPPED

The stored procedure server has not been started, or has been stopped.

STOPPING

A STOP PSERVER command has been issued, but the server has not stopped yet because a stored procedure is executing.

- If a stored procedure is currently running in that server, the name of the stored procedure.
- The authorization ID that is allowed to run the stored procedure. If this field is blank, any user can run the stored procedure.

Note: This command can be issued from the operator console or from ISQL.

GROUP

Indicates that any names that are specified are server group names, rather than individual server names. This enables the database administrator to show information for all servers in a given group with a single command.

name The name of the stored procedure server, or of the stored procedure server group, for which information is to be displayed.

An asterisk (*) in the last position of the name acts as a wildcard. For example, 'ABC*' indicates that the command should be applied to all stored procedure servers (or server groups, if the GROUP keyword is specified) that have names beginning with 'ABC'.

- * Shows information for all stored procedure servers. Note that 'SHOW PSERVER GROUP *' and SHOW PSERVER '*' have the same result - all stored procedure server information will be displayed.

The following are examples of the SHOW PSERVER command.

```
SHOW PSERVER *
SHOW PSERVER SERV1
SHOW PSERVER SERV1*
SHOW PSERVER GROUP GRP1 GRP2
```

The first example displays information for all stored procedure servers for the database. The second statement displays information for SERV1 only. The third shows information for any server whose name starts with 'SERV1'. The last statement shows information for all servers in groups GRP1 and GRP2.

The following example shows the output from the SHOW PSERVER command.

```

show pserver *
Status of stored procedure server(s) 1997-09-21 13:35:49
PSEVER GROUP PROCNAME AUTHID STATUS IMPL
DEFSERV MOD1 STARTED. N
PSERV1 PSERV PMOD1 USER1 STARTED. Y
PSERV2 PSERV PMOD2 USER2 STOPPING Y
PSERV3 PSERV PMOD3 STOPPING Y
ARI0065I Operator command processing is complete.

show pserver group pserv
Status of stored procedure server(s) 1997-09-21 13:35:49
PSEVER GROUP PROCNAME AUTHID STATUS IMPL
PSERV1 PSERV PMOD1 USER1 STARTED. Y
PSERV2 PSERV PMOD2 USER2 STOPPING Y
PSERV3 PSERV PMOD3 STOPPING Y
ARI0065I Operator command processing is complete.

show pserver servyyy
Status of stored procedure server(s) 1997-09-21 13:42:54
ARI4163I Stored procedure server SERVYYY is not defined.
ARI4158E Error in operator command SHOW PSEVER.
Reason Code 6.
PSEVER GROUP PROCNAME AUTHID STATUS IMP
ARI0065I Operator command processing is complete.

```

Figure 76. A Typical SHOW PSEVER Display on a VM Operator Console

SHOW SQLDBGEN

▶▶—SHOW SQLDBGEN—◀◀

The SHOW SQLDBGEN command displays the current database configuration.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show sqldbgen
F4 004
F4 004 The current database definitions are:
F4 004 CUREXTNT=2
F4 004 MAXPOOLS=32
F4 004 MAXEXTNT=64
F4 004 MAXDBSPC=1000
F4 004
F4 004 Non-recoverable storage pools:
F4 004 POOL 2 NOLOG
F4 004
F4 004 Storage pool assignments:
F4 004      Dbextent      Storage Pool
F4 004          1              1
F4 004          2              2
F4 004
F4 004 Current DBSPACES:
F4 004      Type      Pages      Pool
F4 004 PUBLIC      12800      1
F4 004 PUBLIC      2048      1
F4 004 PUBLIC      8192      1
F4 004 PUBLIC      1024      1
F4 004 PUBLIC      512      1
F4 004 PUBLIC      512      1
F4 004 PUBLIC      512      2
F4 004 PUBLIC      512      2
F4 004 PRIVATE     128      2
F4 004 PRIVATE     512      2
F4 004 PRIVATE     128      2
F4 004
F4 004 Internal DBSPACES:
F4 004 INTERNAL 80 1024 1
F4 004
F4 004

```

Figure 77. A Typical SHOW SQLDBGEN Display on a VSE Operator Console

```

show sqldbgen

The current DB definitions are:
CUREXTNT=2
MAXPOOLS=32
MAXEXTNT=64
MAXDBSPC=1000

Non-recoverable storage pools:
POOL 2 NOLOG

Storage pool assignments:
  Dbextent      Storage Pool
    1              1
    2              2

Current DBSPACES:
  Type      Pages      Pool
PUBLIC    12800      1
PUBLIC     2048      1
PUBLIC     8192      1
PUBLIC     1024      1
PUBLIC      512      1
PUBLIC      512      1
PUBLIC      512      2
PUBLIC      512      2
PRIVATE    128       2
PRIVATE     512      2
PRIVATE     128      2

Internal DBSPACES:
INTERNAL   80 1024 1

```

Figure 78. A Typical SHOW SQLDBGEN Display on a VM Operator Console or ISQL Terminal

The SHOW SQLDBGEN columns are defined as follows:

CUREXTNT

The number of dbextents currently defined. This is the current value of CUREXTNT from the database directory.

MAXPOOLS

The maximum number of storage pools as established at database generation time. A storage pool is made up of one or more database extents (DBEXTENTS). This is the value of MAXPOOLS from the resid SQLDBGEN file.

MAXEXTNT

The maximum number of database extents (dbextents) as established at database generation time. Dbextents are the physical media on which the database manager stores the data. This is the value of MAXEXTNT from the resid SQLDBGEN file.

It is possible to reach the maximum physical page limit before reaching the maximum number of dbextents. See “Total number of physical pages” on page 83 for more information. For information on what to do if you reach the maximum physical page limit, refer to the *DB2 Server for VSE & VM Database Administration* manual.

MAXDBSPC

The maximum number of dbspaces as established at database generation time. MAXDBSPC includes public, private, and internal dbspaces. In the resid SQLDBGEN file, this is the value of MAXDBSPC.

Non-recoverable storage pools

A list of storage pool(s) which are nonrecoverable. Existing storage pools which are not listed have been defined with the default LOG option.

Storage pool assignments

The current assignment of dbextents to storage pools.

Current DBSPACES

The current public and private dbspaces, the number of pages defined for each, and to which storage pool each dbspace is assigned.

Internal DBSPACES

The current number of equal size internal dbspaces defined to the database.

SHOW STORAGE



```
▶▶—SHOW STORAGE—▶▶
```

The SHOW STORAGE command displays storage values currently allocated and in use, and free and high storage values for each real agent; as well, it displays total system storage allocation, system high storage values and the user IDs that last caused the high storage marks. The SHOW STORAGE command does not have any parameters.

The SHOW STORAGE command can be used with the RESET HIGHSTOR command to help determine storage required to perform a specific function. By resetting the HIGHSTOR value, performing a function and then invoking the SHOW STORAGE command, you can determine the maximum storage required for that function.

The following examples show the output from the SHOW STORAGE command.


```

show storage
Status of Storage at DATE='1998-09-22' TIME='14:54:09'

AGENT      LOC  ALLOCATED    IN USE      FREE  HIGHSTOR  USERID
OPERATOR   A    4096         0           4096   4096
           B     16         0           16     16
CHECKPT    A   2048         0          2048   2048
           B     16         0           16     16
RECOVERY   A   4096         0          4096   4096
           B     16         0           16     16
TCP/IP     A    512         0           512    512
           B     16         0           16     16
1          A   8192         0          8192   71096  MARISSA
           B     16         0           16     16
2          A   8192         0          8192   8192
           B     16         0           16     16
3          A   8192         0          8192   8192
           B     16         0           16     16
4          A   8192         0          8192   8192
           B     16         0           16     16
5          A   8192         0          8192   8192
           B     16         0           16     16
PROTOTYPE  A  701888      693664     8224   701888
           B  360488     307336     53152  360488
SUMMARY
QUEUES     A   753600     693664     59936  816504
           B  360632     307336     53296  360632
PACKAGES   A    49008      47240      1768   49008
SYSTEM     A   802664      865568
           B    391728      391728
ARI0065I Operator command processing is complete.

```

Figure 79. SHOW STORAGE Display on VM

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 SHOW STORAGE
F4 004 Status of Storage at DATE='1998-09-22'  TIME='14:54:09'
F4 004
F4 004 AGENT      LOC  ALLOCATED      IN USE      FREE      HIGHSTOR  USERID
F4 004 OPERATOR  A    4096          0           4096      4096
F4 004          B    16            0            16        16
F4 004 CHECKPT   A   2048          0           2048      2048
F4 004          B    16            0            16        16
F4 004 RECOVERY  A   4096          0           4096      4096
F4 004          B    16            0            16        16
F4 004 1         A   8192          0           8192      71096  MARISSA
F4 004          B    16            0            16        16
F4 004 2         A   8192          0           8192      8192
F4 004          B    16            0            16        16
F4 004 3         A   8192          0           8192      8192
F4 004          B    16            0            16        16
F4 004 4         A   8192          0           8192      8192
F4 004          B    16            0            16        16
F4 004 5         A   8192          0           8192      8192
F4 004          B    16            0            16        16
F4 004 PROTOTYPE A  701888        693664      8224      701888
F4 004          B  360488        307336      53152     360488
F4 004 SUMMARY
F4 004 QUEUES    A  753088        693664      59424     816504
F4 004          B  360616        307336      53280     360616
F4 004 PACKAGES A   49008         47240       1768      49008
F4 004 SYSTEM   A   802152        865568
F4 004          B   391712        391712
F4 004 ARI0065I Operator command processing is complete.

```

Figure 80. SHOW STORAGE Display on VSE

The SHOW STORAGE display fields are described below.

Header line

indicates the date and time that the SHOW STORAGE command *started* processing.

AGENT

is the agent number or the agent identifier. PROTOTYPE is listed here because it has storage properties similar to an agent.

LOC

is the location of storage, which can reside either above or below the 16MB boundary. Storage that may reside **anywhere** above or below the 16MB boundary is indicated by the letter A. Storage that is restricted to **below** the 16MB boundary is indicated by the letter B.

ALLOCATED

is the current amount of storage set aside for the indicated user or for the system.

IN USE

is the total working storage currently being used by the agent or agents.

FREE

is the total working storage allocated that is not currently being used. It is likely that this free storage is not contiguous.

HIGHSTOR

refers to the maximum amount of storage that has been allocated to this point in time for each indicated user (or for the system). This storage value will reset to the current allocated value when the RESET HIGHSTOR command is entered.

USERID

is the user ID attributed to the agent when the HIGHSTOR value was set.

SUMMARY

is the heading that precedes the summary information for the SHOW STORAGE display. All totals displayed are approximate.

QUEUES

the summary values of all the working storage queues.

Note: Some storage, such as storage used for local variables in SQL/DS code, is not included in these numbers.

PACKAGES

is the summary of the storage queues maintained for packages.

SYSTEM

is the total of all system storage requests. It can contain both direct system storage requests and storage added to the working storage queues.

SHOW SYSTEM

The SHOW SYSTEM command displays much of the data described in the other SHOW commands.

- Date and time are displayed.
- The output of each of the following commands is displayed: SHOW DBEXTENT, SHOW LOG, SHOW CONNECT ACTIVE, and SHOW LOCK MATRIX.

Note that the SHOW SYSTEM command internally issues a SHOW CONNECT ACTIVE command. If you enter the SHOW SYSTEM command from either the DB2 Server for VM operator console or from the VSE system operator console, certain agents are marked forceable. See “Resolving CICS In-Doubt Logical Units of Work” on page 152 for more information about forceable agents.

The following examples show the output from the SHOW SYSTEM command.

```

AR
MSG F4
AR 015 1140I READY
F4 004 ARI0062A SQLDS :
F4-004      Enter a DB2 Server for VSE operator command.
4 show system
F4 004 System state at DATE='01-15-96'  TIME='11:57:35'
F4 004
F4 004 POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
F4 004 NO.      PAGES     PAGES USED  FREE PAGES  RESV PAGES  USED    EXTENTS SOS
F4 004 1      11514     2710       8804       20         23     1
F4 004 2      7467      136        7331       20         1      7
F4 004 FREE  1394139
F4 004 Log Status:
F4 004 Active Log: LOGDSK1
F4 004 Inactive Log: ALTLGD1 - is not archived
F4 004      Log size is 4661248 bytes.
F4 004      Log used is 0 percent 4686 bytes.
F4 004      Log remaining before overflow procedure is 89 percent
F4 004 4190394 bytes.
F4 004      No non-R/O agents began before the last checkpoint.
F4 004      Archiving is not enabled.
F4 004      Log pages remaining before checkpoint is 10.
F4 004 Status of agents:
F4 004      Checkpoint agent is not active.
F4 004      User Agent: 1 User ID: MARISSA is R/O APPL BE01
F4 004      Agent is processing and is in I/O wait.
F4 004      User Agent: 2 User ID: ANDREW is R/O APPL BE13
F4 004      Agent is processing and is in LOCK wait.
F4 004      User Agent: 3 User ID: LAURA is R/O APPL BE10
F4 004      Agent is processing and is in I/O wait.
F4 004      User Agent: 4 User ID: DAVID is R/O APPL BE14
F4 004      Agent is processing and is in LOCK wait.
F4 004      User Agent: 5 User ID: DANIEL is R/O APPL BE12
F4 004      Agent is processing and is in I/O wait.
F4 004      0 agent(s) not connected to an APPL or SUBSYS.
F4 004 Lock Request Block (LRB) and Lock Status:
F4 004 NLRBS  IN USE  FREE  NLRBU  MAX USED BY LUW
F4 004 -----
F4 004      2520    452    2068    1000    554
F4 004      *** THE LOCKWAIT TABLE ***
F4 004      ENTRY = DBSPACE NUMBER ON WHICH THERE IS LOCK CONTENTION
F4 004      The presence of an entry shows
F4 004      the agent requesting the lock and
F4 004      the agent contending for or holding the lock.
F4 004      AGENT          AGENT CONTENDING FOR OR HOLDING THE LOCK
F4 004      REQUESTING
F4 004      LOCK
F4 004          1          2          3          4          5
F4 004          MARISSA  ANDREW  LAURA  DAVID  DANIEL
F4 004 1  MARISSA  .....
F4 004 2  ANDREW  .....11 .....
F4 004 3  LAURA  .....
F4 004 4  DAVID  .....11 .....11 .....
F4 004 5  DANIEL  .....
F4 004
F4 004 ARI0065I Operator command processing is complete.

```

Figure 81. SHOW SYSTEM Display on VSE

```

show system
System state at DATE='01-15-96' TIME='11:57:35'

POOL      TOTAL      NO. OF      NO. OF      NO. OF      %      NO. OF
NO.       PAGES     PAGES USED  FREE PAGES  RESV PAGES  USED  EXTENTS SOS
  1       11514      2710      8804      20      23      1
  2        7467      136      7331      20      1      7
FREE  1394139
Log Status:
Active Log: LOGDSK1
Inactive Log: ALTLGD1 - has not been archived
Log size is 4661248 bytes.
Log used is 0 percent 4686 bytes.
Log remaining before overflow procedure is 89 percent
4190394 bytes.
No non-R/O agents began before the last checkpoint.
Archiving is not enabled.
Log pages remaining before checkpoint is 10.
Status of agents:
Checkpoint agent is not active.
User Agent: 1 User ID: MARISSA is R/O APPL BE01
Agent is processing and is in I/O wait.
User Agent: 2 User ID: ANDREW is R/O APPL BE13
Agent is processing and is in LOCK wait.
User Agent: 3 User ID: LAURA is R/O APPL BE10
Agent is processing and is in I/O wait.
User Agent: 4 User ID: DAVID is R/O APPL BE14
Agent is processing and is in LOCK wait.
User Agent: 5 User ID: DANIEL is R/O APPL BE12
Agent is processing and is in I/O wait.
0 agent(s) not connected to an APPL or SUBSYS.
Lock Request Block (LRB) and Lock Status:
NLRBS  IN USE  FREE  NLRBU  MAX USED BY LUW
-----
  2520   452   2068   1000   554
*** THE LOCKWAIT TABLE ***
ENTRY = DBSPACE NUMBER ON WHICH THERE IS LOCK CONTENTION
The presence of an entry shows
the agent requesting the lock and
the agent contending for or holding the lock.
AGENT          AGENT CONTENDING FOR OR HOLDING THE LOCK
REQUESTING
LOCK
          1          2          3          4          5
          MARISSA  ANDREW  LAURA  DAVID  DANIEL
1  MARISSA  .....
2  ANDREW  .....11 .....
3  LAURA  .....
4  DAVID  .....11 .....11 .....
5  DANIEL  .....

ARI0065I Operator command processing is complete.

```

Figure 82. SHOW SYSTEM Display on VM

SHOW USERS (Valid for VM only)

▶—SHOW USERS—◀

The SHOW USERS command displays a summary of users (DB2 Server for VM applications or ISQL users) connected to the database machine. The operator and ISQL users can enter this command.

SHOW USERS displays the following information:

- The number of users connected to the database machine.

- The number of users who are using an agent structure.

The lines of output have these meanings:

n USER(S) ARE CONNECTED TO SQL/DS

is the number (n) of users connected to the database machine.

n SQL/DS USER(S) ARE ACTIVE

is the number (n) of users who are using an agent structure.

n SQL/DS USER(S) ARE WAITING

is the number (n) of users connected to the database virtual machine and waiting to use an agent structure.

If you are using the VM/ESA operating systems, a VM user can be connected to an application server under one or more different DB2 Server for VM authorization IDs, within separate CMS units of work, with one work unit waiting for another. If a user is waiting for a real agent held by the user's own ID, the SHOW USERS command displays the information like this:

```

2      SQL/DS USER(S) ARE WAITING
      User ID = SQLUSER3  SQL ID = JOHN1
      User ID = SQLUSER3  SQL ID = JOHN2

```

In this situation, you must force one of JOHN's work units to end its work. For information on forcing users to end their work, see "Forcing Users to End Their Work" on page 51.

n SQL/DS USER(S) ARE INACTIVE

is the number (n) of users connected to the database machine, neither using an agent structure nor waiting to use an agent structure.

User ID is the user's VM user ID or user ID in another non-VM environment.

SQL ID is the user's SQL authorization ID.

n SQL/DS AGENTS ARE AVAILABLE

is the difference in value between the value specified by the DB2 Server for VM NCUSERS initialization parameter and the number of users currently using an agent structure.

n SQL/DS USER CONNECTIONS ARE AVAILABLE

is the number (n) of connections available to any new users who want to connect to the database machine.

Note: The SHOW USERS command can be useful during application server shutdown processing (SQLEND) in determining whether there are inactive users who are preventing the application server from ending. The application server does not end (unless you enter SQLEND QUICK) until all users have ended their connections with the database machine. If TCP/IP is enabled, the TCP/IP agent will be included in the SHOW USERS command output as one of the connected users. It is not required to be ended by the user before shutdown. The SQLEND shutdown process will disconnect the TCP/IP agent without requiring the operator's intervention.

The following example shows the output from the SHOW USERS command.

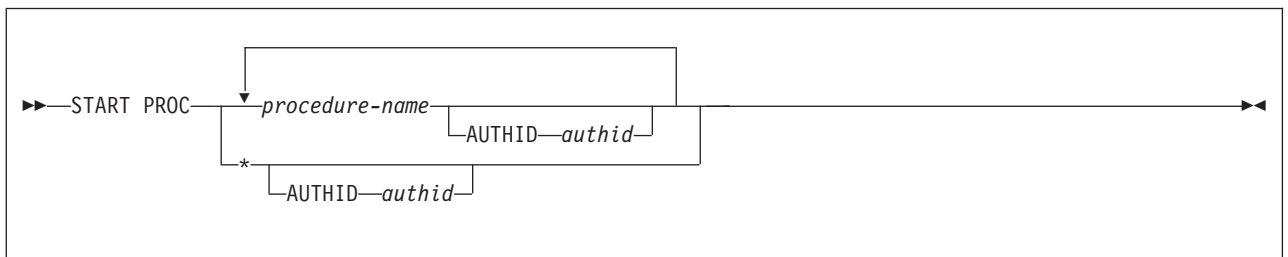
```

show users
Status of Connected Users:                1998-07-24  10:13:41
 4 users are connected to SQL/DS.
 1 SQL/DS users are active.
   User ID: PETERS   SQL ID: SMITH   not processing
 2 SQL/DS users are waiting.
   User ID: DORA    SQL ID: TASK115
   User ID: MICHAEL SQL ID: MIKE2
 1 SQL/DS users are inactive.
   User ID: JENNY   SQL ID: JENNY
 0 SQL/DS agents are available.
94 SQL/DS user connections are available.
ARI0065I Operator command processing is complete.

```

Figure 83. A Typical SHOW USERS Display on a VM Operator Console or ISQL Terminal

START PROC



The START PROC command can be used to:

- Allow the database administrator to restart a stored procedure that was stopped by the STOP PROC command
- Allow the database administrator to refresh a stored procedure after replacing a stored procedure's load module or phase. This is required because the IBM Language Environment loads stored procedures and does not refresh the loaded copy as long as the Language Environment is intact.

The START PROC command starts a stored procedure that was stopped previously. If the stored procedure was not stopped prior to the START PROC command, a message is displayed and no action is taken. In this case, since the procedure is left in STARTED status, the command is considered to be successful.

Upon successful completion of the START PROC command, queued requests for the specified stored procedures are allowed to begin execution. Additionally, the abnormal termination counts for the specified stored procedures are set to zero.

Note: This command can be issued only from the operator console.

procedure-name Identifies the procedure to be started. An asterisk (*) contained in the last position of a procedure name acts as a wildcard. For example, 'ABC*' indicates the command should be applied to all stored procedures whose names begin with 'ABC'.

Message ARI4151I is displayed for each procedure started. If the name specified does not contain a wildcard, and is not found, message ARI4152E is displayed.

Note: *procedure-name* cannot be AUTHID or ACTION, because these are treated as keywords by the START PROC and STOP PROC commands.

- authid* The authorization ID for this stored procedure. This field is used for procedures that can be run only from one authorization ID. Generally, if no authorization ID is specified, only the public version of the stored procedure (that is, the one that does not have an AUTHID associated with it) is started. However, START PROC * is a special case. If START PROC * is issued, all versions of all procedures are started.
- *
- Marks all procedures as eligible to be invoked by the SQL CALL statement.

The following are examples of the START PROC command.

```
START PROC *
START PROC PROC2*
START PROC PROC1
START PROC MYPROC AUTHID USER1
START PROC * AUTHID USER1
START PROC MYPROC AUTHID USER1 YOURPROC AUTHID USER2
```

The first statement above starts all stored procedures. The second starts the public version (that is, the version that does not have an AUTHID associated with it) of all procedures whose names start with 'PROC2'. The third statement starts the public version of one stored procedure, PROC1. The fourth example starts the version of procedure MYPROC that is accessible only by authorization ID USER1. The fifth starts all procedures that are accessible only by authorization ID USER1. The last statement starts the version of MYPROC that is accessible only by USER1 and the version of YOURPROC that is accessible only by USER2.

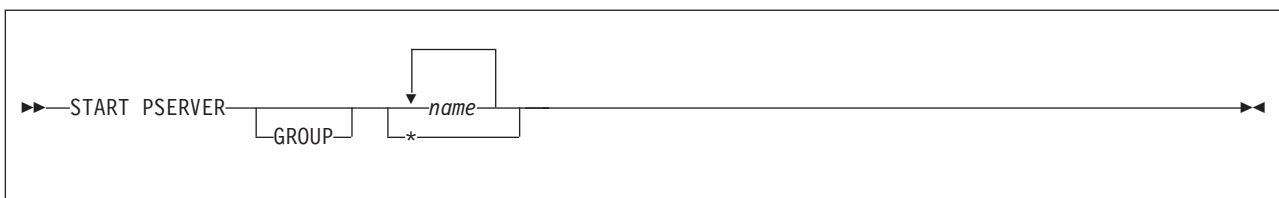
The following example shows the output from the START PROC command.

```
start proc proc* tquery2 yyy
Starting stored procedure(s) 1997-07-21 13:56:22
ARI4151I START PROC successful for PROC1.
ARI4151I START PROC successful for PROC2.
ARI4153I Stored procedure PROC3 is already started.
ARI4151I START PROC successful for PROC3.
ARI4151I START PROC successful for TQUERY2.
ARI4154I Stored procedure YYY is not found.
ARI4152E START PROC failed for YYY.
Reason Code 1.
ARI0065I Operator command processing is complete.

start proc storproc1 authid user1
Starting stored procedure(s) 1997-09-21 14:01:00
ARI4151I START PROC successful for STORPROC1 for AUTHID
USER1.
ARI0065I Operator command processing is complete.
```

Figure 84. A Typical START PROC Display on a VM Operator Console

START PSERVER



The START PSERVER command is used to start a stored procedure server, establish a connection between it and the database machine, and initialize the stored procedure environment.

When the START PSERVER command is issued, the database manager checks the cached information from SYSTEM.SYSPSERVERS for a row in which the value of the column PSERVER matches the name specified on the command. If a row is found, the action taken depends on the status:

- If the status is STARTED, message ARI4164I is issued and no action is taken.
- If the status is STARTING, message ARI4165I is issued, followed by message ARI4167E with reason code 5.
- If the status is STOPPING, message ARI4165I is issued, followed by message ARI4167E with reason code 6.
- If the status is STOPPED, the cache is updated to indicate that the stored procedure server is starting and processing of the command continues.

If the name is not found, message ARI4163I is displayed. In this case, if only one stored procedure server is to be started, the command terminates. If more than one stored procedure server is to be started, the database manager continues with the next one.

Starting a stored procedure server is a two part process. In the first part, when the START PSERVER command is issued, the status of the stored procedure server is updated to STARTING. A storage block is reserved, to be used for the connection between the database manager and the stored procedure server. The second part takes place the first time SQL CALL is processed. If the stored procedure server that is to run the stored procedure has a status of STARTING, the database manager completes the START PSERVER processing prior to sending the SQL CALL statement to the stored procedure server.

Note: If the stored procedure server does not start successfully within the time specified by the PTIMEOUT initialization parameter, message ARI4168I is displayed and the command terminates.

This command can be issued only from the operator console.

GROUP Indicates that any names that are specified are server group names, rather than individual server names. This enables the database administrator to start all servers in a given group with a single command.

name The name of the stored procedure server, or of the stored procedure server group, that is to be started.

An asterisk (*) in the last position of the name acts as a wildcard. For example, 'ABC*' indicates that the command should be applied to all stored procedure servers (or server groups, if the GROUP keyword is specified) that have names beginning with 'ABC'.

Note: *name* cannot be any of the following:

- GROUP
- IMPLICIT
- NOIMPLICIT
- NORMAL
- QUICK

because these are treated as keywords by the START PSERVER and STOP PSERVER commands.

- * Starts all stored procedure servers. Note that 'START PSERVER GROUP *' and 'START PSERVER *' have the same result - all stored procedures servers will be started.

The following are examples of the START PSERVER command.

```
START PSERVER *
START PSERVER SERV1
START PSERVER SERV1*
START PSERVER GROUP GRP1 GRP2
```

The first example starts all stored procedure servers for the database. The second statement starts SERV1 only. The third starts any server whose name starts with 'SERV1'. The last statement starts all servers in groups GRP1 and GRP2.

The following example shows the output from the START PSERVER command.

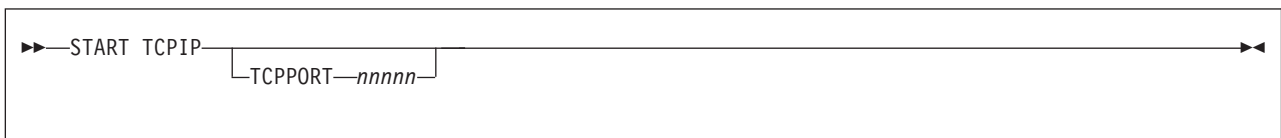
```
start pserver *
Starting stored procedure server(s) 1997-09-21 14:23:06
Starting stored procedure server PSERV1
Starting stored procedure server PSERV2
ARI4164I Stored procedure server DEFSERV
is already started.
ARI4165I START PSERVER cannot be processed for SERYYZ
because it has a status of STARTING.
ARI4167E START PSERVER for SERYYZ failed.
Reason Code 5.
ARI0065I Operator command processing is complete.

start pserver group pserv
Starting stored procedure server(s) 1997-09-21 14:37:40
ARI4165I START PSERVER cannot be processed for PSERV1
because it has a status of STARTING.
ARI4167E START PSERVER for PSERV1 failed.
Reason Code 5.
ARI4165I START PSERVER cannot be processed for PSERV2
because it has a status of STARTING.
ARI4167E START PSERVER for PSERV2 failed.
Reason Code 5.
ARI0065I Operator command processing is complete.

start pserver seryyy seryyx
Starting stored procedure server(s) 1997-09-21 14:37:40
ARI4163I Stored procedure server SERYYY is not defined.
ARI4167E START PSERVER for SERYYY failed.
Reason Code 1.
ARI4166I Unable to acquire storage for a pseudoagent
for server SERYYX.
ARI4167E START PSERVER for SERYYX failed.
Reason Code 3.
ARI0065I Operator command processing is complete.
```

Figure 85. A Typical START PSERVER Display on a VM Operator Console

START TCPIP



The START TCPIP command is used to restart TCP/IP support. Once the TCP/IP support for the application server is initiated, a TCP/IP agent is created to handle all TCP/IP related functions. If any TCP/IP function failure was detected by the

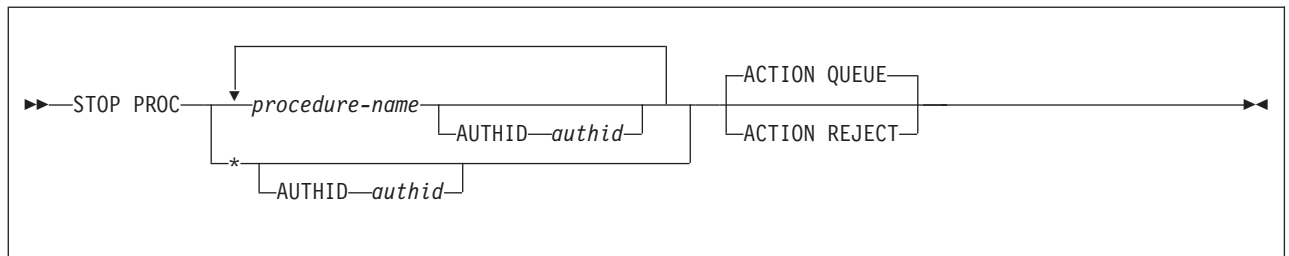
TCP/IP agent, the TCP/IP support for the application server will be disabled. It is possible to restart the TCPIP support for the application server without recycling the application server.

TCPIP Requests to restart TCP/IP support using the port number previously used during database startup.

TCPPORT *nnnnn* Requests to restart TCP/IP support using the port number *nnnnn* specified. *nnnnn* must be a number within 1 and 65535.

Note: For VSE, the same TCP/IP port number can be used among different application servers to initialize TCP/IP support. In this case, the last server using the duplicate port number will become the target server owning the port. All later incoming connect requests using this port number will be connected to this target server. To avoid misdirected connection requests, duplicate port numbers should not be used.

STOP PROC



The STOP PROC command is introduced to allow the database administrator to temporarily stop access to one or more stored procedures. The database administrator can choose to either queue or reject requests to run a stored procedure that has been stopped.

The database administrator may need to stop access to one or more stored procedures for any of the following reasons:

- Stop usage of a stored procedure that is not behaving correctly.
- Add or replace the load module or phase associated with a stored procedure.

When STOP PROC is issued, the database manager looks for the procedure in the cache. If it is found, the status of the procedure is updated, and message ARI4155I is displayed indicating that the command was successful. If the procedure is not found, message ARI4154I is displayed, and then message ARI4157E is displayed, indicating that the command was not successful.

The database manager automatically performs a STOP PROC ACTION REJECT operation on any stored procedure that exceeds the installation-defined maximum abnormal termination count.

The STOP PROC command prevents the database manager from accepting SQL CALL statements for one or more stored procedures. The STOP PROC command does not prevent CALL statements that have already been assigned to a stored procedure server from running.

The effects of the STOP PROC command do not persist after SQLEND has been issued. If the installation wishes to permanently disable a procedure, any of the following actions can be taken:

- Update the LOADMOD column in the SYSTEM.SYSROUTINES table entry, to identify a nonexistent load module or phase.
- Use the DROP PROCEDURE statement to remove the row associated with the stored procedure from the SYSTEM.SYSROUTINES table.
- Rename the load module or phase associated with the stored procedure.
- Delete the load module or phase associated with the stored procedure.

Note: This command is valid only when issued at the operator console.

procedure-name

Identifies the procedure to be stopped. An asterisk (*) contained in the last position of a procedure names acts as a wildcard. For example, 'ABC*' indicates that the command should be applied to all stored procedures whose names begin with 'ABC'.

Note: **procedure-name** cannot be AUTHID or ACTION, because these are treated as keywords by the START PROC and STOP PROC commands.

authid The authorization ID for this stored procedure. This field is used for procedures that can be run only from one authorization ID. Generally, if no authorization ID is specified, only the public version of the stored procedure (that is, the one that does not have an AUTHID associated with it) is stopped. However, STOP PROC * is a special case. If STOP PROC * is issued, all versions of all procedures are stopped.

* Stops all stored procedures.

ACTION

Identifies the action to be taken when an SQL CALL statement is received while the STOP PROC command is in effect. If the STOP PROC command is issued multiple times for a given procedure, the action taken is determined by the ACTION keyword on the most recent STOP PROC command.

QUEUE

Updates the cached information for the procedure with a status of STOP-QUE, which causes the request to be queued until the stored procedure is started via the START PROC command, or until the request exceeds the installation timeout value. This is the default.

REJECT

Updates the cached information for the procedure with a status of STOP-REJ, which causes the request to be rejected.

The following are examples of the STOP PROC command.

```
STOP PROC *
STOP PROC PROC2*
STOP PROC PROC1
STOP PROC PROC1 ACTION REJECT
STOP PROC MYPROC AUTHID USER1
```

The first statement above stops all stored procedures. The second stops the public version (that is, the version that does not have an AUTHID associated with it) for all procedures whose names start with 'PROC2'. The third statement stops the

public version of one stored procedure, PROC1. The fourth example also stops procedure PROC1, but changes the cached status to STOP-REJ, so that further attempts to call this procedure will be rejected rather than queued. The last example stops the procedure MYPROC, which is accessible only by authorization ID USER1.

The following example shows the output from the STOP PROC command.

```

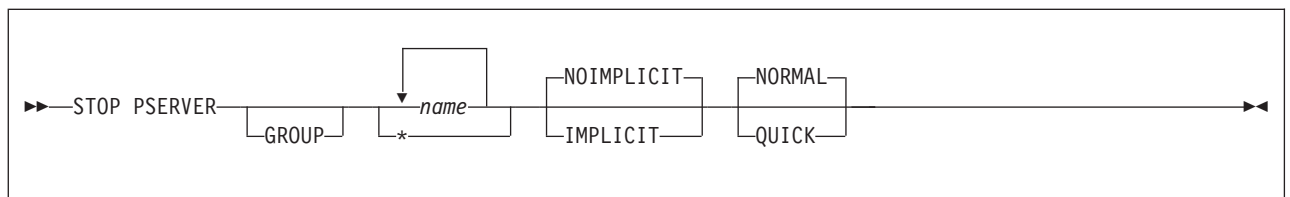
stop proc proc* tquery1 tquery2 authid user2 yyy action queue
Stopping stored procedure(s) 1997-09-21 15:02:04
ARI4155I STOP PROC successful for PROC1.
ARI4156I New calls to stored procedure PROC1
will be QUEUED.
ARI4155I STOP PROC successful for PROC2.
ARI4156I New calls to stored procedure PROC2
will be QUEUED.
ARI4155I STOP PROC successful for PROC3.
ARI4156I New calls to stored procedure PROC3
will be QUEUED.
ARI4159I Stored procedure TQUERY1 is already stopped.
ARI4155I STOP PROC successful for TQUERY1.
ARI4156I New calls to stored procedure TQUERY1
will be QUEUED.
ARI4155I STOP PROC successful for TQUERY2 for AUTHID
USER2.
ARI4156I New calls to stored procedure TQUERY2
with AUTHID USER2 will be QUEUED.
ARI4154I Stored procedure YYY is not found.
ARI4157E STOP PROC failed for YYY.
Reason Code 1.
ARI0065I Operator command processing is complete.

stop proc badproc action reject
Stopping stored procedure(s) 1997-09-21 15:09:07
ARI4159I Stored procedure BADPROC is already stopped.
ARI4155I STOP PROC successful for BADPROC.
ARI4156I New calls to stored procedure BADPROC
will be REJECTED.
ARI0065I Operator command processing is complete.

```

Figure 86. A Typical STOP PROC Display on a VM Operator Console

STOP PSERVER



The STOP PSERVER command is used to stop the handler on a stored procedure server, sever the connection between the stored procedure server and the database machine, and stop the stored procedure server.

When the STOP PSERVER command is issued, the database manager checks the cached information from SYSTEM.SYSPSERVERS for a row in which the value of the column PSERVER matches the name specified on the command. If such a row is not found, message ARI4163I is issued. In this case, if only one stored procedure server is to be stopped, the command terminates. If more than one stored procedure server is to be stopped, the database manager continues with the next one. Note that the stored procedure server does not have to be started for the

STOP PSERVER command to be successful. This can be useful, for example if the database administrator wants to enable implicit starts for a procedure server that was stopped with the NOIMPLICIT option. In this case the database administrator would issue the STOP PSERVER command for that procedure server, specifying IMPLICIT. A STOP PSERVER NORMAL can also be overruled by a STOP PSERVER QUICK. However, a STOP PSERVER QUICK cannot be overruled by a subsequent STOP PSERVER NORMAL.

Note: This command can be issued only from the operator console.

GROUP Indicates that any names that are specified are server group names, rather than individual server names. This enables the database administrator to stop all servers in a given group with a single command.

name The name of the stored procedure server, or of the stored procedure server group, that is to be stopped.

An asterisk (*) in the last position of the name acts as a wildcard. For example, 'ABC*' indicates that the command should be applied to all stored procedure servers (or server groups, if the GROUP keyword is specified) that have names beginning with 'ABC'.

Note: *name* cannot be any of the following:

- GROUP
- IMPLICIT
- NOIMPLICIT
- NORMAL
- QUICK

because these are treated as keywords by the START PSERVER and STOP PSERVER commands.

* Stops all stored procedure servers. Note that 'STOP PSERVER GROUP *' and 'STOP PSERVER *' have the same result - all stored procedures servers will be stopped.

IMPLICIT/NOIMPLICIT

Determines whether the database manager can implicitly activate the stored procedure server. When the database manager is looking for a stored procedure server to use to run a stored procedure, there is a situation in which it will issue the START PSERVER command to start a stored procedure server, which will then be used to run the stored procedure. This is known as an implicit start.

When the STOP PSERVER command is issued, the IMPLICIT/NOIMPLICIT option defaults to NOIMPLICIT. However, any stored procedure server that have not been explicitly stopped since the database manager was started can be implicitly started. In other words, the STOP PSERVER command must be issued to disallow implicit startups of stored procedure servers.

Note that a status of NOIMPLICIT is not maintained after SQLEND. This means that servers that were stopped with the NOIMPLICIT option prior to an SQLEND must be explicitly stopped after the database manager is restarted if implicit startups are to be disallowed.

QUICK/NORMAL

Determines whether a stored procedure that is currently running in the server being stopped should be allowed to finish. Specifying **NORMAL** will allow the procedure to end normally, and after it finishes, processing of the **STOP PSERVER** command will continue. Specifying **QUICK** will terminate the stored procedure immediately. The default is **NORMAL**. In either case, no new stored procedure will be started in the server being stopped.

The following are examples of the **STOP PSERVER** command.

```
STOP PSERVER *
STOP PSERVER SERV1
STOP PSERVER SERV1*
STOP PSERVER GROUP GRP1 GRP2
```

The first example stops all stored procedure servers for the database. The second statement stops **SERV1** only. The third stops any server whose name starts with 'SERV1'. The last statement stops all servers in groups **GRP1** and **GRP2**.

The following example shows the output from the **STOP PSERVER** command.

```
stop pserver pserv1 implicit
Stopping stored procedure server(s) 1997-09-21 15:42:53
ARI4171I Stored procedure server PSERV1
        has been stopped. It can be implicitly started.
ARI0065I Operator command processing is complete.

stop pserver pserv1*
Stopping stored procedure server(s) 1997-09-21 15:44:43
ARI4172I Stored procedure server PSERV1
        is already stopped.
ARI4171I Stored procedure server PSERV1
        has been stopped. It cannot be implicitly started.
ARI4171I Stored procedure server PSERV10
        has been stopped. It cannot be implicitly started.
ARI4173I Stored procedure server PSERV15 is waiting for
        stored procedure PROCX to finish.
Stopping stored procedure server PSERV15
ARI4173I Stored procedure server PSERV16 is waiting for
        stored procedure PROCY to finish.
Stopping stored procedure server PSERV16
ARI0065I Operator command processing is complete.
ARI4171I Stored procedure server PSERV16
        has been stopped. It cannot be implicitly started.
ARI4171I Stored procedure server PSERV15
        has been stopped. It cannot be implicitly started.

stop pserver group group1 group2
Stopping stored procedure server(s) 1997-09-21 16:09:10
ARI4172I Stored procedure server PSERV1
        is already stopped.
ARI4171I Stored procedure server PSERV1
        has been stopped. It cannot be implicitly started.
ARI4171I Stored procedure server PSERV2
        has been stopped. It cannot be implicitly started.
ARI4171I Stored procedure server PSERV3
        has been stopped. It cannot be implicitly started.
ARI0065I Operator command processing is complete.
```

Figure 87. A Typical **STOP PSERVER** Display on a VM Operator Console

Chapter 4. Recovering from System or Media Failures

This chapter contains information on recovering from system and media failures. The first part of this chapter discusses how to restart the application server if the system ends abnormally. The second part, *Resolving CICS In-Doubt Logical Units of Work*, discusses how to resolve logical units of work that have not been committed or rolled back when you restart the application server. CICS in-doubt logical units of work can only occur in a DB2 Server for VM VSE guest sharing environment. The final part of this chapter discusses what to do if you have a DASD failure.

General Recovery

While the application server is processing, serious errors can occur. These errors can cause the operating system, the CICS subsystem, or the application server to end abnormally. The errors can be detected by the following:

- Hardware
- Operating system
- Other subsystems
- Operator
- application server

When an error causes the application server to end abnormally, it displays error messages and reason codes. Also, if the error condition appears to be an internal error, a minidump is sent to the application server console. A minidump is sent to the display device indicated in the DSPLYDEV initialization parameter on a VSE system. You should save all this information for problem determination and refer to the *DB2 Server for VSE & VM Diagnosis Guide and Reference* manual.

First Failure Data Capture support is also available in a VM/ESA system to help in problem determination if the following conditions exist:

- You specify AUTO for the PROTOCOL parameter of the SQLSTART EXEC and the requester connects using DRDA.
- An error occurs in the DRDA portions of the database manager code.

With First Failure Data Capture support, if an error occurs, relevant diagnostic information, control blocks, and data areas are captured at the error detection point in a VM console and sent to the print queue of the application server or requester. For further information on first failure data capture support, see the *DB2 Server for VSE & VM Diagnosis Guide and Reference* manual.

For general recovery, restart the application server the same way you would usually start it. For example, in VSE:

```
// JOB MULTI
// EXEC PROC=ARIS75DB
// EXEC PROC=ARIS75PL
// EXEC ARISQLDS,SIZE=AUTO,PARM='PARMID=WARM1'
```

And in VM,

```
SQLSTART DBNAME(server-name) ID(MYBOOT) PARM(PARMID=WARMSTRT)
```

After you restart the application server, you must resolve any in-doubt logical units of work. In-doubt logical units of work can only occur for VSE guests accessing an application server on VM.

Resolving DRDA 2 In-Doubt Logical Units of Work

A DRDA 2 in-doubt logical unit of work, also known as a distributed unit of work (DUOW), occurs when phase one of the commit processing was completed successfully but phase two did not. A prepare to commit log record for the unit of work was written but the commit log record was not. This situation can occur when a system failure occurs somewhere at the point in time after the prepare to commit has occurred and before the second phase of the commit has occurred.

The system failure could be a DB2 Server for VSE & VM failure, a network failure, a failure on VM CRR for VM or a failure on CICS for VSE, or a failure on the application requester.

When a DUOW is in-doubt, it holds a real agent on the application server. The SHOW CONNECT command would show that a real agent is being held and that it prepared for commit or rollback.

```
show connect
Status of Connected Users          1998-04-07 13:42:03
Checkpoint agent is not active.
User Agent: 1 User-ID: SYSA SQL-ID: VSEMCH12
is prepared for COMMIT or ROLLBACK.
VM ID = VSEMCH12 Coordinator = DBDCCICS Resource Adapter = 0
Transaction= ABC Sigon ID= SYSA Terminal= D080
0 Users are active.
0 Users are waiting.
0 Users are inactive.
2 Agents are available.
45 User connections are available.
ARI0065I Operator command processing is complete.
```

Figure 88. SHOW CONNECT Example of DUOW

To distinguish a DUOW in-doubt agent from a CICS in-doubt agent, look at the Resource Adapter value. If the resource adapter has a value of 254, it is a DUOW in-doubt on VM. If the resource adapter has a value of 255 is a DUOW in-doubt on VSE. If the resource adapter has any other value it is a CICS in-doubt. The value in our example is 254, so this is a VM DUOW. For more information we issue the SHOW INDOUBT command.

```

show indoubt
Status of in-doubt units of work          1997-03-05 13:46:54
TRANID: 26CD User ID: GMERKE
is prepared for COMMIT or ROLLBACK.
LUWID: CAIBMOML.OXNV108.D6BD1D33FA93.0001
EXTNAM: ADHOC.EXE          01CC0001
Requester: DDCS/2 V2.1.1 at BEDROCK
Package: GMERKE.ADHOC      Section: 1
PTC state started: 1997-03-05 13:40:32
Heuristic state started: N/A
Damage: No
ARI0065I Operator command processing is complete.

```

Figure 89. SHOW INDOUBT Example of DUOW

The SHOW INDOUBT command gives more information for the DUOW. CICS in-doubt units of work will not appear in the SHOW INDOUBT display. This example shows that this DUOW has not been resolved yet. This can be determined because the state is prepared for COMMIT or ROLLBACK and the heuristic state started time is N/A and the damage is no.

Normally, in-doubt units of work are resolved automatically by resynchronization recovery. In VM, resynchronization recovery occurs shortly after the application server has been initialized. It is driven by the VM CRR recovery server. In VSE, resynchronization recovery occurs when the DRDA 2 TRUE is enabled for the application server. This happens when an application requester tries to establish a DRDA 2 connection to the application server. If it is successful, there would be no in-doubt agent displayed by the SHOW CONNECT command and there would be no in-doubt units of work displayed by the SHOW INDOUBT command. If it is unsuccessful, the DUOW must be resolved manually.

The logs on the application requester must be examined to determine what action should be taken. The application requester is the coordinator of the DUOW. Most application requesters also have commands to show which units of work are in-doubt and what their status is. DB2 Common Server has the "DB2 LIST INDOUBT TRANSACTIONS" command. DB2 for MVS has the "DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*)" command. The information from the application requester should be used to determine if the DUOW should be committed or rolled back.

The FORCE command is used to resolve the DUOW. The SHOW CONNECT command tells us the in-doubt agent is number 1. If the decision is to commit the transaction, then the FORCE 1 COMMIT command would be issued. Since this is a DUOW in-doubt, you will be prompted to confirm this action. If you are sure, reply 1 for yes and the in-doubt agent will be committed. A subsequent SHOW CONNECT command will show that the agent is free. A subsequent SHOW INDOUBT command will show that the in-doubt was heuristically committed.

```

show indoubt
Status of in-doubt units of work          1997-03-05 14:06:54
TRANID: 26CD User ID: GMERKE
is COMMITTED-H.
LUWID: CAIBMOML.OXNV108.D6BD1D33FA93.0001
EXTNAM: ADHOC.EXE          01CC0001
Requester: DDCS/2 V2.1.1 at BEDROCK
Package: GMERKE.ADHOC      Section: 1
PTC state started: 1997-03-05 13:40:32
Heuristic state started: 1997-03-05 14:05:07
Damage: Unknown
ARI0065I Operator command processing is complete.

```

The heuristic state started time is updated with the time that the FORCE command was performed. The damage is set to unknown. If resynchronization recovery is performed after the DUOW in-doubt was forced, damage may result. If the in-doubt was committed and recovery asked commit to be done, then damage is update to no. Similarly, damage is no if the in-doubt was rolled back and recovery asked rollback to be done. However, damage is yes if the in-doubt was committed and recovery wanted rollback or if the in-doubt was rolled back and recovery wanted commit.

If a DUOW in-doubt was resolved manually, the entry in the SHOW INDOUBT display will remain there until it is removed with the RESET INDOUBT command. The RESET INDOUBT command should not be used until the database administrator is sure that the in-doubt has been correctly resolved at all of the participating sites of the DUOW.

Useful CRR Commands (Valid for VM only)

DB2 for VM uses CRR (Coordinated Resource Recovery) to manage distributed unit of work activity. When an in-doubt unit of work is created in DB2 for VM, the CRR recovery server will **also** have information regarding it. This information can be seen using the "CRR QUERY LUWID" CRR command. If the CRR QUERY LUWID command were to be issued for the in-doubt agent described above, the following information would be displayed on the CRR console:

```

crr query luwid 178
Time: 15:20:45                CRR QUERY LUWID - VMSYSR
Date: 03/06/97                LUNAME - CAIBMOML.OECGW001

LUWID                        Token
1 CAIBMOML.OMXNV108.D6BD1D33FA93.0001 00000178
Name                          Process
2 SQLMACGM                          RESYNCHRONIZATION PENDING
Syncpoint Role                Syncpoint State      Status
INITIATOR CASCADE            COMMITTED
Transaction Tag
3 DB2 FOR VM 5.1.0 PACKAGE: GMERKE.ADHOC

Initiator Name
CAIBMOML.OMXNV108 SQLMACGM
Recovery TPN
.2
'06F2'X
Recovery Token                Index
CAIBMOML.OMXNV108.20B5A76838B5AA31 1 4
Resync Role                   Resync State          Access Userid
RESYNC NEEDED                 RESYNC NEEDED         SQLMACGM

Resources
*LOCAL SQLMACGM
Recovery TPN
SQLMACGM
Recovery Token                Index
6 000026CD                        2 5
Resync Role                   Resync State          Access Userid
8 RESYNC NEEDED                 RESYNC NEEDED 7 SQLMACGM

DMS5BC3065I Operator command processing complete

```

Here are some notes on how this information relates to the information displayed by SHOW INDOUBT at the DB2 for VM application server:

- 1** This is the LUWID of the in-doubt transaction. It should match the LUWID displayed on the SHOW INDOUBT command.

- 2** This is the RESID for the DB2 for VM Application Server. It indicates at which DB2 for VM server the SHOW INDOUBT command should be issued.
- 3** This is a transaction tag set up by DB2 for VM. It consists of the following information:
- Identification of the database (i.e. DB2 for VM).
 - The version, release and modification level of the database (i.e. 5.1.0).
 - The name of the executing package (i.e. PACKAGE: GMERKE.ADHOC). Note that in some cases this can say "PACKAGE: NONE". This means that the no package was active at the time.
- 4** This index number indicates the first of two protected resources displayed by the CRR command. This refers to the protected conversation with the remote requester (and its sync point manager).
- Note:** The index number can be used with the "CRR RESYNC" CRR command to heuristically force this resource from CRR. See *VM/ESA CMS File Pool Planning, Administration, and Operation* for more details.
- 5** This index number indicates the second of two protected resources displayed by the CRR command. This refers to the resource at the DB2 for VM application server itself.
- Note:** The index number can be used with the "CRR RESYNC" CRR command to heuristically force this resource from CRR. See *VM/ESA CMS File Pool Planning, Administration, and Operation (SC24-5751)*, for more details.
- 6** The recovery token is DB2 for VM's internal logical unit of work identifier. This should match the TRANID value displayed by SHOW INDOUBT. If this value is zero, then the unit of work at DB2 for VM was read-only.
- 7** This is the state of the unit of work according to CRR. A value of "RESYNC NEEDED" indicates that resynchronization recovery must still be done. When resynchronization recovery has completed successfully, this value changes to "COMMITTED" or "BACKOUT" depending on what was required.
- 8** This indicates the role that CRR is taking on for this logical unit of work. A value of "RESYNC NEEDED" indicates that resynchronization has started. When resynchronization recovery has completed successfully, this value changes to "FORGET".

When resynchronization recovery has completed successfully at DB2 for VM, the CRR QUERY LUWID command will show information about this unit of work until resynchronization has completed with the requester's sync point manager. That is, when resource for "index 1" of the unit of work (as displayed by the CRR QUERY LUWID command) has been resynchronized, CRR will forget about this luwid.

The following CRR operator command may be used to manage activity at the CRR operator console:

CRR ERASE LU	Erases specified LU name and TPN entries from the CRR log name table
CRR ERASE LUWID	Erases CRR log records for a specified LUWID

	instance, which prevents any further CRR recovery server activity on this LUWID instance
CRR QUERY LOG	Displays the status of the CRR log minidisks
CRR QUERY LOGTABLE	Displays LU names and TPNs in the CRR log name table
CRR QUERY LU	Displays status of logical units of work known to this CRR recovery server and associated with the specified LU name
CRR QUERY LUWID	Displays status of sync point processing and resynchronization processing for an LUWID instance known to this CRR recovery server
CRR RESUME	Restarts the automatic periodic retry of resynchronization for a specified LUWID that was suspended by the CRR SUSPEND command and also bypasses the timed wait interval
CRR RESYNC	Provides a heuristic response for an unavailable protected resource or protected conversation so resynchronization can continue
CRR SUSPEND	Stops the automatic periodic retry of resynchronization for a specified LUWID until the CRR operator enters the CRR RESUME command

These CRR commands are discussed in chapter “CRR Administration” of the *VM/ESA: CMS File Pool Planning, Administration, and Operation* manual.

Resolving CICS In-Doubt Logical Units of Work

A logical unit of work is a sequence of SQL statements. Users control when logical units of work begin and end. They expect all SQL statements within a logical unit of work either to finish successfully as a unit or fail as a unit.

When the system fails, the logical units of work that were in progress at the time of the failure must be resolved. Usually, the application server automatically resolves outstanding logical units of work the next time you start the system.

If a failure occurs when a CICS online user was in the process of ending (committing) a logical unit of work, your intervention may be required. The logical units of work that were being committed at the time of failure are in-doubt logical units of work.

In the CICS subsystem, the CICS restart resynchronization facility automatically resolves in-doubt units of work when you enter the CIRB or CIRA transactions. To activate this facility, you must update the CICS tables to include the resynchronization transactions.

You must manually resolve in-doubt logical units of work if the CICS restart resynchronization facility fails. The procedures are listed below.

If you enter the CIRB or CIRA transaction without resolving the in-doubt logical units of work, the CIRB transaction lists the units that must be resolved, issues diagnostic messages, and ends.

To resolve in-doubt logical units of work, you must enter the SHOW ACTIVE, SHOW CONNECT or SHOW SYSTEM command. (The latter two issue the SHOW ACTIVE command internally.) The SHOW ACTIVE command determines the status of the logical units of work. The agents that you must force are indicated in the SHOW ACTIVE display by the phrase "IS PREPARED FOR COMMIT OR ROLLBACK". For example, in VM:

```
User Agent 7 User ID: JONES is prepared
for commit or rollback
VM ID= VSE1 COORDINATOR= DBDCCICS RESOURCE ADAPTER= 0
TRANSACTION= TPSP CICS SIGNON= PETER TERMINAL= L012
```

And in VSE,

```
F4 004 User Agent 7 User ID: JONES is prepared
F4 004 for commit or rollback
F4 004 COORDINATOR= DBDCCICS RESOURCE ADAPTER= 0
F4 004 TRANSACTION= TPSP CICS SIGNON= PETER TERMINAL= L012
```

The fields in the SHOW ACTIVE display have the following meanings:
COORDINATOR is the CICS subsystem ID.
RESOURCE ADAPTER is the DB2 Server for VSE resource adapter ID.
TRANSACTION is the CICS transaction identifier (TRANSID) for the application.
CICS SIGNON is the CICS signon user ID (if any).
TERMINAL is the CICS terminal ID (if any).

The CICS SIGNON and TERMINAL values may be blank if the data was not available to the application server.

If TRANSACTION=CISQ or if the transaction accessed only the application server, you should commit the logical unit of work. (CISQ is the default CICS transaction identifier for the ISQL facility.) Use the FORCE command, described on page 51. For the above example, you enter:

```
FORCE 7 COMMIT
```

If the transaction accessed multiple resources, you should contact the transaction owner. Otherwise, see the *DB2 Server for VSE System Administration*, or *DB2 Server for VM System Administration* manual for more information on deciding whether to force a commit or roll back.

Application server processing continues when logical units of work are in-doubt, but:

- The resource adapter identified in the SHOW ACTIVE display may not start normal processing for new online applications.
- The locks held by the in-doubt logical units of work are not available to other users.
- The agent structure is not available to other users.

Recovery from Media Failure

Human or machine error could cause the storage media of the log or the database to be unreadable or damaged. For these extreme cases, you must replace the damaged database VSAM file or VM minidisk and restart the application server in recovery mode.

Usually, the database administrator does the media recovery. Replacing a database VSAM file requires knowledge of the physical design of the database. Refer to the *DB2 Server for VSE System Administration* manual for media recovery procedures. Replacing a database minidisk requires knowledge of the physical design of the database. Refer to the *DB2 Server for VM System Administration* manual for media recovery procedures.

After you restore the VSE database, you must ensure coordinated recovery from multiple resource access as discussed in the previous section.

Chapter 5. Tracing the DB2 Server for VSE & VM System

This chapter describes how to start and stop a DB2 Server for VSE & VM trace.

The database manager provides a facility that traces logic and data flow within the VSE partition or VM virtual machine. This trace facility is useful for problem determination and isolation. You can also use it to audit security.

DB2 Server for VSE & VM Tracing

The trace facility lets you trace functions and subcomponents at varying levels of detail. If tracing is active, the database manager writes trace records to an output file, or to memory and then to an output file, whenever execution reaches an internal trace point.

After tracing is complete, you can use the trace formatter supplied with the database manager to select, format, and display or print records from the trace file. Refer to “Formatting DB2 Server for VM Trace Output” on page 166 and “Formatting DB2 Server for VSE Trace Output” on page 168 for descriptions of the trace formatter.

To use tracing, you must:

1. Ensure that your job control to start the application server contains statements that identify the trace output file.
2. Decide what functions and subcomponents you would like to trace, and at what level of detail.
3. Start the trace.

Specifying CMS FILEDEF for a DB2 Server for VM Trace

The trace output file is usually on tape, and is created with IBM standard (EBCDIC) tape labels as a default. You can, however, direct tracing to a CMS file as described later in this chapter. IBM provides an application program, the trace formatter, to select and print trace records from the trace file.

The trace facility requires a CMS FILEDEF command for the trace output file. A FILEDEF command for standard label tape processing is provided in the SQLSTART EXEC. You can override the supplied FILEDEF command by issuing your own FILEDEF command before invoking the SQLSTART EXEC. You must specify the PERM option on your CMS FILEDEF commands if you are running single user mode and the program is written in a language other than Assembler.

The trace file can be a standard labeled tape or unlabeled tape or a CMS file. If the trace tape file is to be multivolume, you must use standard labels. The SQLSTART EXEC does not enter a CMS LABELDEF command. The SQLSTART EXEC specifies standard label (SL) processing on the FILEDEF command. If you want to change the tape label processing, enter appropriate CMS FILEDEF and LABELDEF commands before you enter SQLSTART. You must ensure that the ddname for both the FILEDEF and LABELDEF is ARITRAC. For more information on DB2 Server for VM tape support, refer to the *DB2 Server for VM System Administration* manual. For more information about tape label processing in CMS, refer to the *VM/ESA: CMS User's Guide* manual.

Specifying Job Control for a DB2 Server for VSE Trace

The output of the trace can be directed to either tape or disk.

To direct the output to tape, include a TLBL statement in your job control for generating a trace. Omit any ASSGN statement for the trace tape unit. The database manager dynamically assigns and unassigns the trace tape unit.

The database manager creates a trace output file on tape with IBM* standard (EBCDIC) tape labels. A block size of 4096 bytes is used.

An example of a TLBL statement for a trace output file is as follows:

```
// TLBL ARITRAC,'TRACE.FILE2'
```

Note: The TLBL statement must have the file name parameter specified as ARITRAC. All other parameters are optional.

The trace tape is never rewound except at end-of-volume.

To direct the output to disk, you need a DLBL, an EXTENT, and an ASSGN statement. If the file is managed by the VSE/VSAM Space Management for SAM Feature, the EXTENT statement may be optional, and the ASSGN statement is not applicable. In any case, the file name on the DLBL statement must be ARITRAC.

When the database manager is tracing to DASD, it uses a block size of 4088 bytes for the trace file.

The following is an example of the required job control for a trace output DASD SAM file (not managed by VSAM).

```
// DLBL ARITRAC,'TRACE.FILE1'  
// EXTENT ,VSER01,1,0,301,120  
// ASSGN SYS007,195
```

Notes:

1. The DLBL file name must be ARITRAC.
2. The DASD allocation (301,120) is 4 cylinders on 3350 volume VSER01 starting at cylinder 10. The volume is on disk drive 195.
3. If you run the tracing more than once, you have to specify a new DASD allocation each time unless you want to write over the old file. (You should also specify a different file-id.)
4. This example uses the default symbolic unit for DB2 Server for VSE DASD output (SYS007).

If you do not want to use the default symbolic unit (SYS007), you must specify the desired symbolic unit as the first EXTENT parameter and in the ASSGN statement. If you are also using the DB2 Server for VSE & VM accounting facility with output to DASD, either the trace or the accounting file must be directed to a symbolic unit other than SYS007.

The following is an example of the job control required for a trace output DASD SAM file that is managed by the VSE/VSAM Space Management for SAM Feature.

```
// DLBL ARITRAC, 'TRACE.FILE1'0, VSAM, DISP=(OLD,KEEP),          C
                        RECORDS=1000, RECSIZE=4088
// EXTENT ,VSER01
```

Note: Unlike standalone SAM, the VSE/VSAM Space Management for the SAM feature lets you extend the file. It also automatically provides secondary space allocation should you exceed the primary allocation. You can monitor the actual space usage in the file by using the VSAM Access Method Services.

1. The DLBL file name must be ARITRAC.
2. The DLBL parameter “VSAM” indicates that this is a VSAM-managed file.
3. The example assumes you wish to implicitly define the file to VSAM the first time that the file is accessed.
4. DISP=(OLD,KEEP) indicates that the file is not to be reset at OPEN time (OLD), and is not to be deleted at CLOSE time (KEEP). This allows you to implicitly define the file the first time the file is used. It also lets you extend the file (add records to it) in subsequent runs of the application server.
5. VSAM uses the RECORDS and RECSIZE parameters when it is implicitly defining the files. The parameters help VSAM determine how much primary and secondary space to allocate for the file.
In this example, the primary allocation is about 4 million bytes (1000 x 4088). The secondary allocation is about 20% of the primary allocation. You should always set RECSIZE to 4088 because the database manager uses VB records with a block size of 4088 when tracing to disk.
6. Note that VSE/VSAM allows up to 15 secondary allocations to the file if the primary allocation is filled.
7. An EXTENT statement with the volume serial number (VSER01 in this example) is normally required for implicit define.
8. No ASSGN statement is required for VSE/VSAM-managed files.
9. This example omits the “DLBL CAT=” parameter. The example assumes that the file is in the VSAM job catalog.
10. The VSAM catalog must own sufficient unallocated space on the specified volume to satisfy the space allocation requirements for this file.

Tracing in Multiple User Mode

In multiple user mode, you can start tracing using the following:

- The TRACCONV, TRACDBSS, TRACDRRM, TRACDSC, TRACRDS, TRACSTG, and TRACWUM initialization parameters
- The TRACE operator command.

Note: If you specify the TRACDRRM or TRACWUM initialization parameters, you receive trace output only if you have users who are operating under the DRDA protocol.

The initialization parameters allow the later parts of initialization to be traced and do not require any operator command entry. The operator command TRACE ON allows tracing to be started at any time after initialization is complete. It also has options to trace only specified agents or a specified authorization ID.

Tracing can be stopped either by allowing the system to end (normally or abnormally) or by issuing the operator command TRACE OFF.

Tracing in Single User Mode

In single user mode, tracing can be started only through the TRACCONV, TRACDBSS, TRACDSC, TRACRDS, TRACSTG initialization parameters. Tracing is stopped only when the application server ends (normally or abnormally).

Tracing to a CMS File

The trace facility, as a default, directs its output to tape. You can, however, direct trace output to memory or to a CMS file. For more information on directing trace output to memory, see “Starting a Trace” on page 160 or Appendix A, “DB2 Server for VSE & VM Initialization Parameters,” on page 179.

You can direct trace output to a CMS file because the database manager uses the CMS simulation of OS QSAM to process the trace file, and CMS OS QSAM is device independent.

To trace to a CMS file, use the following CMS FILEDEF command before starting the application server.

```
FILEDEF ARITRAC DISK filename filetype filemode (PERM
```

Notes:

1. The ddname for the trace file must be ARITRAC, as shown.
2. The file mode letter can be any properly accessed CMS minidisk.
3. The file mode number must be 4. This is required because the trace record format is CMS OS QSAM variable blocked (RECFM VB).
4. You can always specify the PERM option. It is required if you are running the application server in single user mode, and the application program is written in a language other than the Assembler language.

After issuing the above CMS FILEDEF command, you can start the application server. As usual, depending on whether you are in single or multiple user mode, you can start tracing through the initialization parameters or through the TRACE operator command.

If you start tracing through the TRACCONV, TRACDBSS, TRACDRRM, TRACDSC, TRACRDS, TRACSTG, and TRACWUM initialization parameters, the application server prompts for the cuu of the trace tape. When tracing to a CMS file, specify in the cuu parameter the virtual device number of the CMS minidisk containing the trace file.

Similarly, if initiating tracing with the TRACE operator command, specify the virtual device number of the trace minidisk for the cuu parameter of the TRACE ON command.

Special Considerations for Multiple Activations of Tracing

If you do the following:

- Activate tracing (using the TRACE ON command or the TRACCONV, TRACDBSS, TRACDRRM, TRACDSC, TRACRDS, TRACSTG, and TRACWUM initialization parameters).
- Terminate tracing (enter the TRACE OFF command)
- Reactivate tracing (enter the TRACE ON command).

the trace CMS output file only contains the trace output of the second or last activation of tracing. This can be avoided in either of two ways:

- When you end tracing with the intention of restarting it, enter the TRACE OFF NOCLOSE command. Note that when the application server ends, it closes the trace file.
- Specify the option DISP MOD on the ARITRAC FILEDEF command. This causes subsequent activations of tracing to add the output to the end of the existing trace output file.

If the trace fills the CMS minidisk on which the trace file resides, the following occurs:

- The CMS system issues a message indicating that the CMS minidisk is full. This message has a DMS prefix.
- Error message ARI0081E (with REASON=12 and CODE=0) is issued.
- Normal operation continues with no trace output.
- In multiple user mode, the operator can enter the SQLEND operator command. This is necessary only if you do not want the application server to run without trace output.
- When the CMS minidisk frequently fills, you should use a larger minidisk or one with more free space for trace output. You may also consider directing the trace output to memory using the TRACEBUF parameter, refer to Appendix A, "DB2 Server for VSE & VM Initialization Parameters," on page 179.

When the application server is running with tracing active, and the system ends (normally or abnormally), the application server displays message ARI0099I if the trace file is successfully closed. The same situation results if the TRACE OFF command is entered. If this message is not displayed, the file has not been closed and some trace data has been lost.

Note: It is possible to enable tracing but produce no trace output, for example, when you trace only a specified authorization ID. In this situation, the file is opened and closed successfully, but no CMS file is created.

Tracing the DB2 Server for VSE Resource Adapter

See the *DB2 Server for VSE & VM Database Administration* manual for information on how to trace the VSE resource adapters.

Choosing the Functions to Be Traced

You can trace functions and subcomponents within the following components:

Data Conversion (CONV)

This component performs numeric data conversion and returns data to

user host variables when the DRDA protocol is used. It is also responsible for performing data conversion on character and graphic data regardless of the protocol in use.

Database Storage Subsystem (DBSS)

This component is the storage manager. It provides locking mechanisms to ensure that multiple concurrent users do not interfere with each other. It also handles recovery, database I/O, and sorting.

Distributed Relational Resource Manager (DRRM)

This component interprets the data streams used in DRDA protocol. It builds (generates) DRDA data streams from DB2 Server for VSE & VM internal format and it translates (parses) DRDA data streams back into DB2 Server for VSE & VM internal format.

Data System Control (DSC)

This component enables communication. It controls the initialization and termination, and acts as the supervisor of the database machine or partition.

Relational Data System (RDS)

This component supports the relational model of data. It allows users to see data as though it exists in tables containing rows and columns. It also translates all SQL statements into specific tasks for the Database Storage Subsystem.

Storage (STG)

This component will enable tracing of both system and working storage used by DB2 Server for VSE & VM on the application server side.

Work Unit Manager (WUM)

This component controls the flow of DDM requests and replies when the DRDA protocol is used. It is the interface between the DRDA component DRRM and the non-DRDA components RDS and DSC.

Starting a Trace

In multiple user mode, you can start a trace by using either the TRACE operator command or the TRACCONV, TRACDBSS, TRACRDS, TRACWUM, TRACDRRM, TRACSTG, or TRACDSC initialization parameters. In single user mode, you can use only the TRACCONV, TRACDBSS, TRACRDS, TRACDSC and TRACSTG initialization parameters. In single user mode, the TRACWUM and TRACDRRM parameters are ignored, even if you specify them. You cannot enter the TRACE operator command in single user mode.

This section describes how to start a trace using the TRACE operator command. Appendix A, "DB2 Server for VSE & VM Initialization Parameters," on page 179 describes the trace initialization parameters.

Both methods of starting a trace have their advantages. The initialization parameters allow part of the initialization to be traced and do not require any operator command entry. The operator command TRACE ON allows tracing to be started at any time after initialization is complete. It also has options to trace only specific agents or authorization IDs.

In VM, the SQLSTART EXEC issues the following CMS FILEDEF command to define the default trace output file:

```
FILEDEF ARITRAC TAP2 SL (BLOCK 4096 NOCHANGE
```

If you want to use different FILEDEF options, enter a CMS FILEDEF command before invoking the SQLSTART EXEC.

Note: The ddname in the FILEDEF command statement must be ARITRAC. If your tape volume is unlabeled, use only the TAPn and NL parameters. The initialization ends abruptly if end of volume is reached on an unlabeled output tape file.

If the tape volumes have the standard labels required for multivolume output, use the TAPn and SL parameters. If you require multivolume output, omit the FILEDEF LEAVE and NOEOV parameters. The LEAVE parameter suppresses rewinding before OPEN processing and after CLOSE processing.

For standard label tapes, the CMS LABELDEF command is optional. When you use the CMS LABELDEF command, you must enter it with the ddname ARITRAC parameter prior to invoking the SQLSTART EXEC.

The trace output file can be a tape or a CMS file. You can use both IBM standard tape labels and unlabeled tapes. Tape rewinding is controlled by CMS FILEDEF command parameters. The CMS FILEDEF command supplied by the SQLSTART EXEC assumes tape output with the use of IBM standard tape labels. It also rewinds the tape.

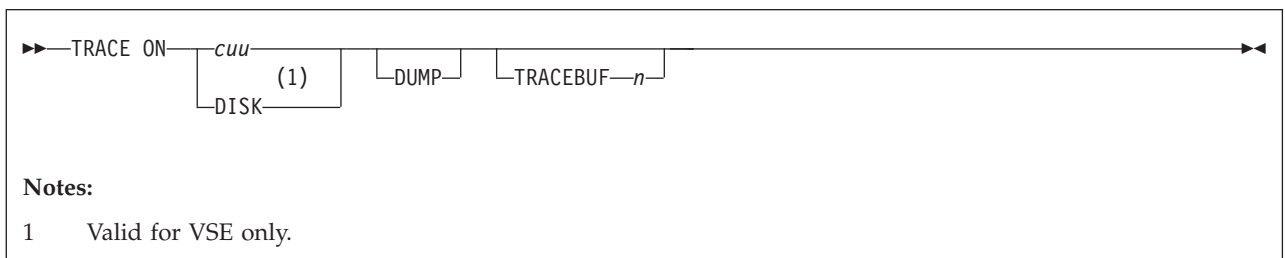
In VSE, to run the trace facility, include statements for the trace output file in your job control that starts the application server. The statements you need vary depending on whether you want the trace output on tape or on disk:

- When you trace to tape, you need a TLBL statement for the trace output file. Your TLBL statement must have the file name parameter specified as ARITRAC. All other TLBL parameters are optional. Omit any ASSGN statement for the trace tape unit. The application server dynamically assigns and unassigns the trace tape unit.

Trace output tape files are created with IBM Standard (EBCDIC) Tape Labels. The trace tape is never rewound except at end-of-volume. You must rewind the tape manually or by using job control statements.

- When you trace to disk, you need a DLBL, an EXTENT, and an ASSGN statement. The file name specified on the DLBL statement must be ARITRAC. If the file is managed by the VSE/VSAM Space Management for SAM Feature, the EXTENT statement is optional and the ASSGN statement is not applicable.

To activate a trace in multiple user mode, enter the TRACE operator command:



ON

indicates trace activation. If trace is already active, you must first enter the TRACE OFF command before you can enter the TRACE ON command.

cuu

is the virtual address of the trace device used for trace output. In VM this is specified in the ARITRAC FILEDEF command. The default address is 182. If you are tracing to tape, a tape volume must be mounted and ready on the tape unit. If you are tracing to a CMS file, enter the virtual device number of the minidisk.

DISK (Valid for VSE only)

indicates that trace output is to a DASD SAM file.

Note: You must specify either *cuu* or DISK.

DUMP

causes a snapshot dump of the VSE partition or VM virtual machine at a particular trace point. If you specify this option, you are prompted for further information as described on page 164.

TRACEBUF *n*

is an optional parameter that specifies the amount of memory, in kilobytes, that you want to allocate for storing trace records. A trace buffer *n* kilobytes in size is created. This buffer holds the last *n* kilobytes of trace information in memory until either a TRACE OFF is issued or the application server ends (normally with an SQLEND, or abnormally). The number of kilobytes, *n*, can be any integer in the range 0 to 99999. If you specify a value of 0, the trace is written directly to the trace output file. When you enter either TRACE OFF or SQLEND, you may choose not to create the trace output file by including the TRCPURGE parameter. This parameter instructs the database manager to purge the contents of the trace buffer instead of writing it to the output file.

A suggested size for the trace buffer is 100 kilobytes or more. It is recommended that you increase the Virtual Machine memory or virtual storage size of the database partition by the size of trace buffer before you start a trace in memory. See item 5 on page 165 in the following TRACE command sequence.

After you enter the TRACE command, the application server requests further information by prompting.

Note: When the prompting messages appear on the DB2 Server for VM operator terminal, the display status area displays "RUNNING" instead of "VM READ". This occurs because the application server continues to service DB2 Server for VM users while it is waiting for the operator reply.

The application server prompts you for the following information:

1. You are prompted to specify the DB2 Server for VSE & VM authorization ID, agent number, or * (for ALL) to indicate the authorization ID or agent structures that you want traced.

For traces, the operator agent is agent 1, the checkpoint agent is agent 2, and the ready/recovery agent is agent 3. The first general purpose agent is agent 4, the second is agent 5, and so on.

The authorization ID for the operator agent is OPERATOR. The authorization ID for the checkpoint agent is CHECKPT. The authorization ID for the ready/recovery agent is RECOVERY. The authorization ID for a general purpose agent is the DB2 Server for VSE & VM authorization ID of the connected user.

2. You are prompted by message ARI0087D to specify whether you want to trace the Data Conversion (CONV), Database Storage Subsystem (DBSS), Relational

Data System (RDS), Work Unit Manager (WUM), Distributed Relational Resource Manager (DRRM), Storage (STG), Data System Control (DSC), or all of these components. The CONV component performs CCSID and numeric conversion. The RDS component is a compiler and translator for the SQL language. The DBSS component controls and accesses the application server. The DSC component controls communications and agent handling. The STG component controls DB2 Server for VSE & VM system and working storage on the application server side. The WUM and DRRM components support the DB2 Server for VM DRDA protocol. You will not be prompted for WUM or DRRM if PROTOCOL=SQLDS.

- For DBSS, RDS, DRRM, and DSC tracing, the system prompts you to specify the list of functions and subcomponents and their trace levels (level 0, 1, or 2). The CONV, WUM, and STG components have no subcomponents; therefore, the system prompts you only for trace levels. For CONV and WUM, if you request tracing at level 0, no tracing is done. If you request tracing at level 1, tracing displays only module entry and exit points. If you request tracing at level 2, tracing displays module entry, report, and exit points. If you request tracing at any level other than 0, 1, or 2, it is an error. For STG, if you request tracing at level 0, no tracing is done. If you request tracing level 1, tracing displays storage trace values that may include amount requested/freed, the storage pointer, and the return code. STG does not have a level 2 trace. For each function or subcomponent, enter the name of the component or function and the level number as follows:

```
component1 level-number1 [component2 level-number2] ...
```

For example: Request tracing at level 1 for LUW and level 2 for DC.

```
LUW 1 DC 2
```

Leave at least one blank between the component name and the level number. The DBSS functions and subcomponents that you can supply in response to message ARI0090D are:

```
ENTRY      (DBSS entry calls)
EXIT       (Returns from DBSS entry calls)
LOG        (Log and Recovery Management)
LOCK       (Lock Management)
LUW        (Logical Unit of Work Management)
DC         (Data Control)
DM         (Data Manipulation)
STOR       (Storage Management)
SORT       (DBSS Sort)
INDEX      (DBSS Index)
STAT       (DBSS Update Statistics)
*          (For all of the above)
```

DBSS has unique subcomponents, ENTRY and EXIT, which provide tracing for DBSS calls before entering or after exiting the module.

ENTRY and EXIT tracing at level 1 display only DBSS Call entry or exit points. Tracing at level 2 displays the base control information and auxiliary structures, depending on the DBSS opcode.

The RDS functions and subcomponents you can supply in response to message ARI0088D are:

EXEC	(Executives)
PA	(Parser)
OPT	(Optimizer)
AG	(Access Generator)
INT	(Interpreter and Authorization)
AU	(Security Audit Trace)
SG	(Statement Generator)
*	(For all of the above)

For information on using the security audit trace facility, refer to the *DB2 Server for VSE & VM Database Administration* manual.

The DSC functions and subcomponents you can supply in response to message ARI00142D are:

AGENT	(Agent handling)
COM	(Communications)
*	(For all of the above)

Tracing the AGENT subcomponent in DB2 Server for VM provides a data stream trace for the application server. You can use it to isolate a DRDA data stream trace. To produce a data stream trace for the application requester, refer to the *DB2 Server for VSE & VM Database Administration* manual.

The DRRM functions and subcomponents that you can supply in response to message ARI0140D are:

DICT	(DDM/FD:OCA Dictionary and FD:OCA descriptors and data)
GEN	(DDM Generator)
PARSE	(DDM Parser)
RDIIN	(RDIIN Manager)
*	(For all of the above)

The CONV component has no subcomponents, so in response to message ARI0160D, you supply only trace levels.

The STG component has no subcomponents, so in response to message ARI0162D, you supply only trace levels.

The WUM component has no subcomponents, so in response to message ARI0141D, you supply only trace levels.

4. If you specify DUMP for the TRACE command, you are prompted to specify a CONV, DBSS, RDS, DRRM, WUM, STG, or DSC trace-point number. This causes a snapshot dump of the database partition machine to occur the first time the specified trace point is activated. Snapshot dumps do not occur with subsequent activation of the trace point.

The snapshot dump is generated by the CP DUMP command, or the VSE PDUMP macro.

Note: The function or subcomponent containing the trace point must be activated for tracing to cause activation of the snapshot dump. Also, the type of dump is controlled by the initialization parameter DUMPTYPE. This parameter has three possible values: P, F, or N. F is the default. If

you specify P, a partial dump of everything except the DB2 Server for VSE & VM code occurs. If you specify F, a full machine dump occurs. If you specify N, no dump occurs.

In DB2 Server for VM, partial virtual machine dumps are sent to the virtual printer and full virtual machine dumps are sent to the virtual reader. Refer to the *DB2 Server for VSE & VM Diagnosis Guide and Reference* manual for dump handling instructions.

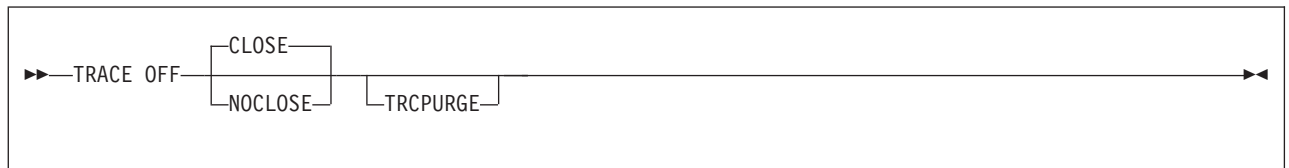
5. If you specify TRACEBUF for the TRACE ON command and there is sufficient free storage available to allocate a trace buffer, you will receive a message ARI0154I indicating the amount of contiguous memory available. If the amount of storage *n* kilobytes specified by the TRACEBUF parameter is not available, then the maximum size of the currently available contiguous storage is displayed in error message ARI0155E and the trace will not be started. You can reenter the TRACE ON command for a different size of trace buffer or not select the trace buffer.

Note: The amount of free storage fluctuates during execution of the DB2 Server for VSE & VM program. If a substantial portion is used by the trace buffer there is a higher probability that an agent may be rolled back or the database system may **shutdown** because of the lack of storage.

For traces started by initialization parameters you cannot trace only a selected authorization ID or agent number or dump at a specified trace point.

Stopping a Trace

In single user mode, the trace ends when the application server ends. (Remember, you cannot enter operator commands in single user mode.) To deactivate a trace in multiple user mode, enter the following operator command:



OFF
indicates trace deactivation.

CLOSE
specifies that the trace file is to be closed. In DB2 Server for VSE it also specifies that the trace file is to be unassigned. The default is CLOSE. If you specify the CLOSE parameter, the trace file will be overwritten when issuing a subsequent TRACE ON command.

NOCLOSE
specifies that the trace file is to be left open for further tracing into the same file with a subsequent TRACE ON.

If you are tracing to tape, specify either CLOSE or NOCLOSE and then TRACE ON to create multiple trace output files.

In DB2 Server for VM you must specify the LEAVE option on the ARITRAC FILEDEF command.

Note: You can use the LEAVE option to process these files sequentially when you are formatting trace output.

TRCPURGE

Specify this optional parameter if you want to purge the contents of the trace buffer when the TRACE OFF command is processed. In this instance an output trace file will not be created for the trace buffer. Otherwise the contents of the trace buffer, if it exists, will be flushed out to the trace output file. The TRCPURGE parameter is ignored if there is no trace buffer in use when TRACE OFF is issued.

If you end normally, the trace file is closed automatically. If the application server is ended abnormally, the file may not be closed successfully. A message is displayed whenever the trace file is closed successfully.

If the trace file is a tape and is not closed successfully, you can still try to format it. Information near the end of the trace might not be processed, however, and the trace formatting utility will probably terminate abnormally. You can also try to write a tape mark on the trace tape before you attempt to format it. (The trace tape is not automatically rewound.)

Formatting DB2 Server for VM Trace Output

The trace formatter is a DB2 Server for VSE & VM-supplied program that transforms the information on the trace tape into a printed report. You control the trace formatter by supplying control statements as input.

Note: You should not attempt to use the trace formatter in the CMS Subset environment.

The trace formatter is invoked through an EXEC called SQLTRFMT. To invoke the SQLTRFMT EXEC, enter SQLTRFMT.

```
▶▶—SQLTRFMT—◀◀
```

The SQLTRFMT EXEC issues the following CMS command to define the trace formatter's default input file:

```
FILEDEF ARITRAC TAP2 SL (NOCHANGE
```

Notes:

1. The SQLTRFMT EXEC does not issue a CMS LABELDEF command.
2. The ARITRAC FILEDEF command used for the formatter must correspond to the ARITRAC FILEDEF command used for the trace itself. If you enter your own trace file ARITRAC FILEDEF command for the trace, you must enter the same FILEDEF command before invoking the SQLTRFMT EXEC.
3. If you want the additional standard label checking provided by the CMS LABELDEF command, you must enter the LABELDEF command before invoking the SQLTRFMT EXEC. The LABELDEF file name, *ddname*, must be ARITRAC.

4. The trace formatter does not require the application server to be running. You must, however, have read access to the production minidisk to run the trace formatter.
5. The trace formatter cannot run in CMS DOS mode. If you run the trace formatter in CMS DOS mode, the CMS session ends abnormally.

If you are tracing to tape, you must ensure that the trace tape is mounted on virtual device number 182 or the virtual device number specified in your (optional) FILEDEF command, before invoking the SQLTRFMT EXEC.

If you direct the trace output to a CMS file, you must enter a CMS FILEDEF command for the file before invoking the SQLTRFMT EXEC. Use the same FILEDEF that you entered before invoking the SQLSTART EXEC to start the trace. Refer to “Tracing to a CMS File” on page 158 for use of the FILEDEF command.

Before invoking the SQLTRFMT EXEC, run the SQLINIT EXEC, if you have not already done so. This ensures that the trace formatter can use the resource adapter to generate messages. For more information on the SQLINIT EXEC and resource adapter tracing, refer to the *DB2 Server for VSE & VM Diagnosis Guide and Reference manual*.

During its processing, the SQLTRFMT EXEC invokes XEDIT to edit a card-image CMS file called SQLTRFMT SQLTRACE A1. The first time you issue the SQLTRFMT EXEC, you can see that the file is empty. The SQLTRFMT EXEC creates it on your A-disk. You should place the trace formatter control statements in this file. The trace formatter control statements are explained in the section “Trace Formatter Control Statements” on page 170.

When you are finished entering the control statements, use the PF3 key to file the results and continue processing, or you can use the XEDIT command FILE. The SQLTRFMT SQLTRACE is then stored on your A-disk and is used as the SYSIN file for the trace formatter program.

The SQLTRFMT EXEC then prompts as to whether you want to direct the trace output to the virtual printer or to a CMS file. If you choose to print the data, the SQLTRFMT EXEC prints to the virtual printer 00E. If you want, you can enter CP SPOOL and TAG commands before invoking the SQLTRFMT EXEC to direct the output to a particular class or device. If you want the output directed to a CMS file, the SQLTRFMT EXEC prompts you for a file name, file type, and as an option, file mode. If you do not specify a file mode, the trace formatter uses A1.

The trace formatting program can generate considerable output. You should ensure that there is enough space on your A-disk or other specified minidisk before directing trace formatter output to a CMS file.

Note: The output file contains 121-character records. The first character is an ANSI carriage control character. If you route your output to a CMS file and later want to print the file, use the CC option on the PRINT command to indicate that the carriage control characters are present.

After you specify where you want the formatted output to go, the SQLTRFMT EXEC invokes the program that formats the trace data. Upon completion, the SQLTRFMT EXEC returns control to CMS. If any control statements are in error, the trace formatter program prints messages in the output CMS file or to the virtual printer (whichever you selected). The SQLTRFMT EXEC displays a message that indicates whether it ran successfully or encountered an error.

The SQLTRFMT SQLTRACE A1 is not erased upon completion. On subsequent runs, the SQLTRFMT EXEC invokes XEDIT for this file and lets you modify it.

Formatting DB2 Server for VSE Trace Output

You control the trace formatter by supplying control statements that identify the trace output file. The trace output file can be on tape or disk. If the file is on disk it might be managed by the VSE/VSAM Space Management for SAM Feature, or it might be a standalone SAM file.

Following is an example of job control to invoke the trace formatter. This example contains a DUMPALL parameter. DUMPALL causes the trace formatter to display on SYSLST all the trace records from the trace file. In the example, the tape is mounted on the virtual address 181, and the file ID is "FORMATTER". The ASSGN statement for a trace tape file must always specify SYS004 as the symbolic unit.

```
// JOB TRACEJOB
// TLBL ARITRAC,'FORMATTER'
// ASSGN SYS004,181
// EXEC ARIMTRA,SIZE=AUTO
DUMPALL
/*
/ &
```

Your job control statements will differ when the trace file is on disk. If your trace output file is on disk, specify the trace formatter DISK control statement. If you are using both DISK and DUMPALL, DISK must precede DUMPALL.

Following are three examples of starting the trace formatter. The examples show how to run the trace formatter when the trace output is:

- A tape file
- A standalone SAM DASD file
- A SAM DASD file managed by the VSE/VSAM Space Management for SAM feature.

The formatter examples work with the corresponding DB2 Server for VSE trace examples shown in "Specifying Job Control for a DB2 Server for VSE Trace" on page 156.

The following is an example of the job control required to invoke the trace formatter when the trace output file is on tape.

```
// JOB RUN TRACE FORMATTER
// TLBL ARITRAC,file-id    <-- File-id of trace tape (optional)
// ASSGN SYS004,cuu        Address of tape unit
// EXEC ARIMTRA,SIZE=AUTO
    •
    control statements for trace formatter
    •
/*
/ &
```

Notes:

1. The tape should be mounted on the physical device specified by cuu before running the job.
2. The tape file-id must be the same file-id as was specified on the TLBL statement when the tape was created.
3. The trace formatter requires at least 65K (115K if the trace file is a VSAM-managed file) of virtual storage to run.

The following is an example of the job control required to run the trace formatter when the trace output file resides on DASD and is not VSAM-managed.

```
// JOB RUN TRACE FORMATTER
// DLBL ARITRAC,'TRACE.FILE1'
// EXTENT ,VSER01,1,0,301,120
// ASSGN SYS006,195
// EXEC ARIMTRA,SIZE=AUTO
DISK
  .
  other control statements for trace formatter
  .
/*
/ &
```

Notes:

1. The DLBL file name must be ARITRAC.
2. In this example, the DB2 Server for VSE default symbolic unit for DASD input is used (SYS006).
3. DISK is a trace formatter control statement. DISK is required whenever the trace output file is on DASD.
4. The trace formatter requires at least 65K (115K if the trace file is a VSAM-managed file) of virtual storage to run.

The following is an example of the job control required to run the trace formatter when the trace output file resides on DASD and is managed by the VSE/VSAM Space Management for SAM Feature.

```
// JOB RUN TRACE FORMATTER
// DLBL ARITRAC,'TRACE.FILE1',0,VSAM,DISP=(,DELETE)
// EXEC ARIMTRA,SIZE=AUTO
DISK
  .
  other control statements for trace formatter
  .
/*
/ &
```

Notes:

1. The file name on the DLBL statement must be ARITRAC.
2. The DLBL parameter "VSAM" indicates that this is a VSAM-managed file.
3. DISP=(,DELETE) indicates that when the file is closed it is to be deleted from the VSAM catalog. If you want to keep the file, specify DISP=(,KEEP).
4. This example omits the DLBL "CAT=" parameter. It assumes that the trace output file is in the VSAM job catalog.

5. DISK is a trace formatter control statement. DISK is required whenever the trace output file is on DASD.
6. The trace formatter requires at least 65K (115K if the trace file is a VSAM-managed file) of virtual storage to run.

COMP and SUBCOMP have been updated for DSC.

Trace Formatter Control Statements

Control statements for the trace formatter select the trace records to be printed. The control statements are identified by these keywords:

AGENTNO	HEADER
COMP	RETCODE
DATE	SUBCOMP
DBNAME	TIME
DISK (VSE Only)	TRACENO
DUMPALL	USERID
EXTLUWID	

In general, each keyword takes one or more parameters. Each parameter is separated by one or more blanks. Do not use commas to separate the parameters. Each control statement can contain only one keyword, in columns 1 to 71 inclusive, with no continuations. You can supply the control statements in any order. Do not place blank control statements in the input to the trace formatter in DB2 Server for VSE; and do not place blank records in the control statement file, SQLTRFMT SQLTRACE in DB2 Server for VM.

The purpose and syntax of each keyword is described as follows:

AGENTNO nn

defines the ordinal numbers of the agent structures for which trace records are to be formatted. An agent number consists of up to 2 digits. Up to six agent numbers can be specified on the AGENTNO control statement, in any order, separated by one or more blanks.

Certain agent numbers are fixed:

The operator agent is always agent 1.

The checkpoint agent is always agent 2.

In single user mode, agent 3 is the only user agent. In multiple user mode, agent 3 is the ready/recovery agent, and higher numbers are user agents.

If TCP/IP support is active, agent 4 is the TCP/IP agent, and higher numbers are user agents. If TCP/IP support is not active, agent 4 and higher numbers are user agents.

If the AGENTNO keyword is omitted, agent number values are not considered in choosing trace records to be formatted.

COMP CONV

COMP DBSS

COMP DSC

COMP DRRM

COMP RDS

COMP STG

COMP WUM

defines the DB2 Server for VSE or DB2 Server for VM component for which trace records are to be formatted. Only one component can be chosen.

If the COMP keyword is omitted, the component that the trace record describes is not considered in choosing trace records to be formatted. The COMP keyword should not be used with the SUBCOMP keyword.

DATE mm/dd/yy

DATE mm/dd/yy mm/dd/yy

defines the date or date interval for which trace records are to be formatted. If only one date is specified, trace records created on that date are formatted. If two dates are specified, trace records created in that interval are formatted. The dates must be specified in month/day/year notation. Each portion of a date must be specified as two digits; that is, a leading zero must be entered where needed, for example, DATE 01/01/90.

If the date statement is omitted, the date is not considered in choosing trace records to be formatted.

Notes:

1. Year numbers less than 43 are assumed to be in the 21st century.
2. A date range which crosses the century boundary cannot be used and the trace formatter must be run twice; once for the desired range in each century.

DBNAME *server_name*

defines the server name for which trace records are to be formatted. A server name consists of up to 18 characters or numbers. Up to six server names can be specified on the DBNAME control statement, separated by one or more blanks.

If you omit the server name, server name values are not considered when you choose records to be formatted.

DISK (Valid for VSE only)

informs the trace formatter that the file generated by the trace is on DASD. You must specify this keyword when the trace file is on DASD. If you specify both the DISK and the DUMPALL keywords, the DISK keyword must precede the DUMPALL keyword.

If the DISK keyword is omitted, the trace input file is assumed to be a tape file.

DUMPALL

requests that all trace object records be formatted. This keyword takes no parameters. This keyword cannot be combined with any other keywords except with DISK.

EXTLUWID N *nnnnnnnn*

Nnnnnnnn is the eight character SNA NETID of the external logical unit of work for which you want to format trace records. The first character (N) is always required. If you specify the first character only, trace points for all connected SNA NETIDs which begin with that character will be returned. Additional characters up to a total of 35 can be added from left to right so that you can format records for specific SNA NETIDs only.

HEADER

specifies that only the header object of the trace output record is to be formatted. Module names, return codes, and variable data are not displayed. This keyword takes no parameters.

RETCODE P

RETCODE N

RETCODE *

defines the types of return codes for which trace records are to be formatted. Only one character can be specified for the RETCODE parameter. The choices are:

- P Positive (Nonzero) Return Codes Only
- N Negative Return Codes Only
- * All Nonzero Return Codes

If the RETCODE keyword is chosen, only trace point records with return codes of the specified value are chosen for formatting. If the RETCODE keyword is omitted, the return codes are not considered in choosing trace records to be formatted.

SUBCOMP *subcomponent*

defines the subcomponents of DBSS, RDS, DRRM, RA, or DSC, for which trace records are to be formatted. The allowable codes are:

For DBSS:

- DC Data Control
- DM Data Manipulation
- ENTRY DBSS Entry Calls
- EXIT Returns to RDS from DBSS Entry Calls
- INDEX Index
- LOCK Lock Management
- LOG Log/Recovery Management
- LUW Logical Unit of Work Management
- SORT Sort
- STAT Update Statistics
- STOR Storage Management

For RDS:

- AG Access Generator
- EXEC Executive
- INT Interpreter and Authorization
- OPT Optimizer
- PA Parser
- AU Security Audit Trace
- SG Statement Generator

For DRRM:

- DICT DDM/FD:OCA Dictionary and FD:OCA descriptors and data
- GEN DDM Generator
- PARSE DDM Parser
- RDIIN RDIIN Manager

For DSC:

- AGENT Agent handling (data stream trace)
- COM Communications

For RA:

- RA Resource Adapter Control Flow
- COM Communications

The SUBCOMP keyword enables you to list the specific subcomponents to be traced. Subcomponents can be specified in any order, each separated by one or more blanks. Up to eight subcomponents can be specified. If the SUBCOMP keyword is omitted, subcomponents are not considered in choosing trace records to be formatted.

The COMP keyword should not be used with the SUBCOMP keyword.

TIME hh:mm:ss hh:mm:ss

defines the time interval for which trace records are to be formatted. The hour:minute:second time specification requires using 24-hour clock notation. Each portion of a time must be entered as two digits; that is, a leading zero must be entered where needed, for example, TIME 09:05:00 14:04:59.

Specifying a time interval that passes through midnight must be done in two different runs of the trace formatter.

It is possible to specify only one time with the TIME keyword. For example, TIME 12:00:00 specifies that only the trace records created during that second of time be formatted.

If TIME is omitted, the time value is not considered in choosing trace records to be formatted.

TRACENO nnnnn

defines the trace point numbers for which trace records are to be formatted. A trace point number consists of up to 5 digits. Up to six trace point numbers can be specified on the TRACENO control statement, in any order, separated by one or more blanks. Trace points are assigned to specific places in the components or modules that make up the DB2 Server for VSE & VM system.

If the TRACENO keyword is omitted, trace point numbers are not considered in choosing trace records to be formatted.

USERID *auth_id*

defines the authorization IDs for which trace records are to be formatted. An authorization ID consists of up to 8 letters or numbers. Up to six authorization IDs can be specified on the USERID control statement, separated by one or more blanks.

Certain authorization IDs are fixed:

The operator agent (agent 1) is always OPERATOR.

The checkpoint agent (agent 2) is always CHECKPT.

The ready/recovery agent (agent 3 in multiple user mode) is always RECOVERY.

The TCP/IP agent (agent 4 if it exists) is always TCP/IP.

If the USERID keyword is omitted, authorization ID values are not considered in choosing trace records to be formatted.

The following example demonstrates the use of trace formatter keywords. Note the following characteristics:

```
Component: RDS component
Subcomponent: Parser
Trace Numbers: 4400, 4401, 4402
Date of Creation: March 11, 1988
Time of Creation: Between 9:12:00 AM and 1:12:00 PM
Agent Number: 4
User: JOHNDOE
```

To print a listing from the trace file for all the trace records that have these characteristics, the input control statements to the trace formatter will be as follows:

```
TRACENO 4400 4401 4402
DATE 03/11/88
TIME 09:12:00 13:12:00
AGENTNO 4
USERID JOHNDOE
```

Note: The control statements can be placed in any order. Because the trace-point numbers are known, the COMP and SUBCOMP keywords are not required.

Format of DB2 Server for VSE & VM Trace Output

Each time a trace point is encountered in a function or subcomponent that is activated for trace and the agent or authorization ID is active for trace, trace-point output is produced.

The first printed line is the Trace Header and it has the following format:

TRACEPOINT=nnnn

identifies the trace point number

DBSS_OP=nn

RDS_OP=nn

DSC_OP=nn...

identifies trace point as DBSS, DSC, or RDS components and gives the current DBSS, DSC, or RDS OPCODE being executed. Special OPCODE numbers are assigned to DBSS non-OPCODE functions such as checkpoint, archive, and warm start, among others.

While other components may be listed on this line, their opcodes will always be zero.

The RDS OPCODEs come from the RDIIN control block on external calls to RDS in RDIIN field RDICTYPE. RDS places the OPCODE in field RDAOPCOD in the RDAREA control block for trace and for problem determination.

The DBSS OPCODEs, excluding special OPCODEs as described in the following paragraphs, originate from DBSS interface (DBSI) calls as the OPCODE parameter. These DBSS OPCODEs are placed in field YT1OPCOD in the YTABLE1 control block for trace and for problem determination.

Certain DBSS functions are executed without formal DBSI calls to the agent that executes that function. In YT1OPCOD, DBSS sets special pseudo-OPCODEs to cover a number of these situations as follows:

OPCODE=99:

DBSS performs warm-start DB2 Server for VSE & VM initialization as a result of the initialization parameter STARTUP=W.

OPCODE=98:

DBSS performs database generation and initialization as a result of the DB2 Server for VSE & VM initialization parameter STARTUP=C.

OPCODE=97:

DBSS is initialized and restores the database from an archive tape as a result of the DB2 Server for VSE & VM initialization parameter STARTUP=R.

OPCODE=96:

DBSS is initialized and adds new dbspaces to the database as a result of the DB2 Server for VSE & VM initialization parameter STARTUP=S.

OPCODE=95:

DBSS is initialized and adds new dbextents to the database as a result of the DB2 Server for VSE & VM initialization parameter STARTUP=E.

OPCODE=94:

DBSS is initialized and redefines or formats the log data sets (COLDLOG) as a result of the DB2 Server for VSE & VM initialization parameter STARTUP=L.

OPCODE=93:

DBSS performs an DB2 Server for VSE & VM checkpoint or checkpoint and archive in the checkpoint agent (agent 2).

OPCODE=92:

DBSS performs an asynchronous ROLLBACK or COMMIT of a LUW. This can be caused by conditions such as deadlock and the FORCE operator command.

OPCODE=91:

DBSS executes, in the operator agent, a DB2 Server for VSE & VM operator command from the DB2 Server for VSE & VM operator.

USERID=ccccccc

identifies the DB2 Server for VSE & VM authorization ID.

For STG trace points, agent 0, the *userid* field will be set to PROTOTYP. PROTOTYP is not an authorized ID, but is set to help identify that the storage operation is from prototype storage and not from general purpose agent storage.

The operator agent, agent 1, always has the authorization ID OPERATOR. The DB2 Server for VSE & VM initialization occurs under that authorization ID.

The checkpoint agent, agent 2, always has the authorization ID CHECKPT.

The ready/recovery agent is agent 3, and can be used in multiple user mode only. Agent 3 always has the authorization ID RECOVERY.

The TCP/IP agent is 4 if TCP/IP support is active. It has the authorization ID TCPIP.

COMP=compname

identifies the name of the subcomponent or function being traced. This is the same as the name specified through the TRACE ON operator command.

AGENT=nn

nn is the ordinal of the DB2 Server for VSE & VM agent structure. The operator is always agent 1. The checkpoint is always agent 2. In single user mode, agent 3 is the only user agent. In multiple user mode, agent 3 is always the ready/recovery agent. In multiple user mode, agents 4 through *n* are the user agents, where *n* is the NCUSERS parameter value plus three.

Note: For STG tracing, some trace points will have agent=0 for the PROTOTYP.

If TCP/IP support is active, agent 4 is always the TCP/IP agent and then agents 5 through *n* are the user agents, where *n* is the NCUSERS parameter value plus four.

DATE=mm/dd/yy

identifies the date of trace point activation.

DBNAME=ccccccc

is the server name being traced. DBNAME appears only in the trace point header when tracing RA, DRRM or CONV from the application requester.

TIME=hh:mm:ss

indicates the time of trace point activation. Time is adjusted to local time as specified by VSE Job Control through the ZONE JCS, or the VM SYSTIME

system generation macro. The time is in 24 hour notation. Thus 2:00 p.m. is 14:00:00. The time zone adjustment is made at Trace Format and Print time.

[LUW_ID='hhhhhhh'X]

is the DBSS Logical Unit of Work ID. This output line appears for DBSS trace points only.

[EXTLUWID='NETID.LUNAME.LUWID_INSTANCE.LUWID_SEQUENCE_NUM']

is the Logical Unit of Work ID for the DRDA protocol.

NETID

The SNA network associated with the connection. It can be one to eight characters long.

LUNAME

The logical unit associated with the connection. In VM, a logical unit is a gateway. It can be one to eight characters long.

LUWID_INSTANCE

A 12 character timestamp in hex.

LUWID_SEQUENCE_NUM

The LUWID sequence number 0001.

For both level 1 and 2 tracing, except for DBSS entry and DBSS exit trace points, the trace header is always followed by:

- If module entry trace point
MOD_CALLED='entry point name'
- If module exit trace point
MOD_RETURNED='module name' (followed by)
RETCODE=---n...n
(present if module passed a return code)
- If neither module entry nor module exit (middle of some module)
MOD_REPORT='module name'

For both level 1 and 2 tracing for DBSS entry trace points, the trace header is always followed by:

DBSS ENTRY: L_OPCODE='DBSS-opcode-name'

For both level 1 and 2 tracing for DBSS exit trace points, the trace header is always followed by:

DBSS EXIT: L_OPCODE='DBSS-opcode-name'
RETCODE=---n...n
(RETCODE is the DBSS Return Code)

For all level two trace points and for components having only 1 level of tracing, processing and debugging variables are displayed after the above information. Variables are displayed in the general form:

- L_varname=value-or-string
or
- G_varname=value-or-string

The L or G prefix indicates that the variable is only locally addressable to the issuing module (L), or is globally addressable through control blocks to all modules (G).

Varname should be the name of the data item (simple entity, structure or substructure) by which the issuing module addresses the data item.

Value-or-string is of the form:

- [-]n...n if displaying decimal data (binary internally)
- 'c...c' if displaying (unconverted) character data
- 'h...h'X if displaying hex data (binary internally)
- L/G_varname=HEXADECIMAL DUMP: (line 1)
 - 0000 hhhhhhhh (up to 12 hex words) (line 2)
 - 0030 hhhhhhhh (up to 12 hex words) (line 3 if needed).

More lines are included as needed for up to 32 kilobytes of hex data. For lines 2 to n, the first 4 characters are the hex offset of displayed hex dump data (for example, 0 for the first line, 30 for the second line). The rest of the line is hex output displayed in groups of 8 hex characters (each a binary word) with 12 groups per line (less only if end of output data reached).

Trace output may also contain printable character strings that are trace point dependent and clarify processing being performed or the data being displayed.

Isolating Problems

You can use either the ISQLMAP or the ISQLTRACE commands to isolate problems within the ISQL facility.

ISQLMAP

The ISQL facility contains its own load module map, which is invoked by the command:

```
ISQLMAP
```

ISQL Dumps in DB2 Server for VSE

Use of CICS dumps in problem diagnosis is presented in the *CICS Problem Determination Guide*

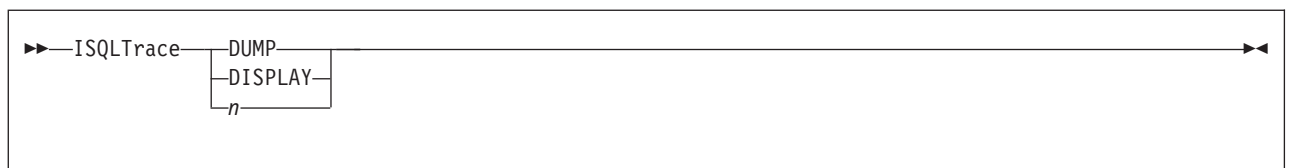
ISQL In-Core Trace

The ISQL facility lets you trace activity within ISQL. Items that are traced include:

- Calls to and returns from ISQL modules
- SQL return codes
- ISQL messages.

The trace is always active. It stores a limited amount of information internally in a trace table.

The initial trace table holds 50 entries. After 50 entries are made, the table wraps and only the last 50 entries to the table are kept. To change the size of the internal table and select how trace information is to be displayed, issue the ISQLTRACE command.



DUMP

creates an unformatted storage dump hardcopy of the trace table.

In a VSE system, the ISQL facility issues a CICS dump to the CICS dump data set. You should use your installation's usual job for printing the dump data set.

In a VM system, the ISQL facility issues a CP DUMP command. The dump uses the printer with the lowest virtual address. The application server issues a CP CLOSE PRINT to close the print file. This also closes any other open print files.

DISPLAY

displays the formatted trace table on the display. If more than 50 (or *n*) entries have been made into the trace table, the entries wrap. Only the last 50 (or *n*) entries in the table are displayed. The entries are displayed last in, first out, in the reverse of the order they were put in the table. Table 2 shows the internal storage layout of these entries in DB2 Server for VM.

- n* to change the size of the trace table. Replace *n* with the number of trace entries that are to be contained in the trace table. *n* must be a number from 50 to 1000.

Table 2. Internal Storage of the ISQL Trace Table on VM

TRMOD CHAR(8)	TRCMOD CHAR(8)	TRR15 FIXED(31)	TRLBL FIXED(15)	TRCODE CHAR(1)	TRCHAR CHAR(5)
calling- returning module	called module	address or return code	2-byte label number	C-calls or R-return	5-bytes char mod dep info
For Example:					
ARIISTR ARIISQL1	ARIISQL1 (hex 0)	0048A808 00000004	0002 0001	C R	SEL EXITS

TRMOD

is the module issuing the call or return.

TRCMOD

is the module being called.

TRR15

holds the value of register 15 at the time of the entry, which is the address of the called module or the return code.

TRLBL

is the number associated with the label of the CALL or RETURN:

A label of TRLC7 would have a TRLBL number = 7, and would be the 7th CALL issued from the module.

A label of TRLR3 would be the third RETURN issued from the module, and would have a TRLBL number = 3.

TRCODE

C indicates a CALL label, R indicates a RETURN label.

TRCHAR

contains module-dependent information. The information saved here varies for each module. It is a maximum of 5 bytes.

The following is an example output of the ISQLTRACE DISPLAY command. The addresses, label numbers, and character information are not shown in the example.


```

ARI7181I ISQL trace table display.
      First entry: 2B36F0 Current entry: 2B3C50 Last entry: 2B3D10
ARIIPSQ CALLED ARIITKN,R15= 443E60,TRLBL= 0001,TRCHAR= TKN1
      RETURN FROM ARIITKN,R15= 000000,TRLBL= 0003,TRCHAR=
ARIIPSQ CALLED ARIITKN,R15= 443E60,TRLBL= 0015,TRCHAR= TKN2
      RETURN FROM ARIITKN,R15= 000000,TRLBL= 0003,TRCHAR=
ARIIPSQ CALLED ARIIPQY,R15= 44C8B0,TRLBL= 0018,TRCHAR= SEL
ARIIPQY CALLED ARIICNV,R15= 458860,TRLBL= 0001,TRCHAR= CONS
      RETURN FROM ARIICNV,R15= 000010,TRLBL= 0001,TRCHAR= DECC
ARIIPQY CALLED ARIIGM ,R15= 44220E,TRLBL= 0031,TRCHAR= SQLDA
:

```

Figure 90. A Typical Example ISQLTRACE DISPLAY Output

Appendix A. DB2 Server for VSE & VM Initialization Parameters

This section describes the parameters you can specify when starting the application server. For a complete description of the initialization parameters, and information on initialization parameter values, talk to your system administrator or see the *DB2 Server for VSE System Administration*, or *DB2 Server for VM System Administration* manual.

Common Parameters

The following table contains initialization parameters common to both the VSE environment and the VM environment. Initialization parameters specific to each system follow.

Parameter	Default	Description
ALTLOG=N Y	N	<p>Alternate Logging</p> <p>This 1-character field indicates whether an inactive log exists in the system. Y indicates an inactive log is present; N indicates no inactive log has been defined.</p> <p>The inactive log will only be used when LOGMODE=L. When ARCHPCT is reached, an attempt will be made to switch to the inactive log rather than take a log archive.</p> <p>This parameter is used only when STARTUP=L or STARTUP=C is specified.</p> <p>Note: The DB2 Server for VM EXECs always supply the ALTLOG parameter for you. You should not specify the ALTLOG parameter directly.</p>

Parameter	Default	Description
ARCHPCT= <i>n</i>	80	<p>Archive Percent</p> <p><i>n</i> is a decimal value indicating percentage. When the log is <i>n</i> percent full, the database manager begins an archive process, prompting the operator to mount the required tape. The <i>n</i> can be any number from 10 to 99.</p> <p>This parameter is used only when LOGMODE=A or when LOGMODE=L is specified. When LOGMODE=A, a database archive is started when the ARCHPCT value is reached. When LOGMODE=L and ALTLOG=N, a log archive is started instead of a database archive. When LOGMODE=L and ALTLOG=Y, an attempt is made to switch to an inactive log disk. For optimum use of the archive function, the value of ARCHPCT should be less than the value of SLOGCUSH.</p>
CHARNAME= <i>name</i>	<p>After installation: INTERNATIONAL</p> <p>After migration: ENGLISH</p>	<p>Character Set Name</p> <p>This 1- to 18-character field specifies the name of the character set that the database manager uses to define the system language. The name you specify must correspond to the name of a character set that has been defined in the NAME column of the SYSTEM.SYSCHARSETS table and the CHARNAME column of the SYSTEM.SYSCCSIDS table.</p> <p>The value of the CHARNAME parameter specified in the SQLSTART EXEC or startup jcl determines the values of the CHARNAME, CCSIDSBCS, CCSIDMIXED and CCSIDGRAPHIC rows in the SYSTEM.SYSOPTIONS table. It also updates the CCSID and SUBTYPE attributes for the character columns of the system catalog tables. If CHARNAME is not specified, the CHARNAME value is determined from the CCSID value of the CNAME column in the SYSTEM.SYSCOLUMNS table. For information on selecting a CHARNAME and the related performance considerations, see the <i>DB2 Server for VSE System Administration</i>, or <i>DB2 Server for VM System Administration</i> manual.</p>
CHKINTVL= <i>n</i>	10	<p>Checkpoint Interval</p> <p><i>n</i> is a decimal value that defines when the database manager is to automatically take a checkpoint. A checkpoint is taken after every <i>n</i> log pages have been written. <i>n</i> can be any number from 1 to 99,999,999.</p>
DBSPNUM= <i>n</i>	1	<p><i>n</i> indicates the dbspace number on which release empty pages processing will be performed. <i>n</i> must be an acquired dbspace number in the database or *.</p> <p>DBSPNUM is only valid when SYSMODE=S and STARTUP=P.</p> <p>Note: The DB2 Server for VM EXECs always supply the DBSPNUM parameter for you. You should not specify the DBSPNUM parameter directly.</p>

Parameter	Default	Description
DISPBIAS= <i>n</i>	7	Dispatcher Bias <i>n</i> is a decimal value from 1 to 10. The DISPBIAS parameter controls how the DB2 Server for VSE & VM dispatcher operates. When the DISPBIAS parameter is set closer to 10, the dispatcher operates more like a priority dispatcher. When the DISPBIAS parameter is set closer to 1, the dispatcher operates more like a round robin dispatcher. Refer to the <i>DB2 Server for VSE System Administration</i> , or <i>DB2 Server for VM System Administration</i> manual for more information on the DISPBIAS parameter.
DSPSTATS= <i>nm</i>	00	This 2 character field allows values 0, 1 and 2 for option 1 and values 0 and 1 for option 2. If the first option is 1, checkpoint performance data is displayed on the console in the form of messages every time a checkpoint occurs. If the first option is 2, checkpoint performance detailed data is displayed on the console in the form of messages every time a checkpoint occurs. If the second option is 1, the COUNTER * command will be internally invoked when the database manager is shutdown. This displays the counter value for the database manager at the last possible moment. If VMDSS is activated, the COUNTER POOL * command is also issued.
DUALLOG=N Y	N	Dual Logging This 1-character field indicates whether the database manager is to maintain an identical copy of the active log and, if ALTLOG=Y, a backup copy of the alternate log. Y indicates dual logging; N indicates single logging. This parameter is used only when STARTUP=L or STARTUP=C is specified. Note: The DB2 Server for VM EXECs always supply the DUALLOG parameter for you. You should not specify the DUALLOG parameter directly.
DUMPTYPE=P F N	F	Dump Type This 1-character field indicates the type of dump to be taken if the application server ends abnormally. F indicates a full dump. For VM, this is the entire virtual machine. For VSE, this is the entire partition. P indicates a partial dump, where everything is dumped except DB2 Server for VSE & VM code. (DB2 Server for VSE & VM control blocks are included in the dump.) N indicates no dump. No dump is taken if a DB2 Server for VSE & VM limit error or a hardware error occurs or if the termination is because of a user specification error. ATTENTION: IBM recommends that you always use DUMPTYPE=F, as most problems cannot be resolved with a partial dump. This will help eliminate the need for recreating problems just to get a full dump. Note: In single user mode, DUMPTYPE=P includes the application code in the dump.

Parameter	Default	Description
EXTEND=Y N	N	<p data-bbox="817 220 1059 249">Filtered Log Recovery</p> <p data-bbox="817 275 1436 478">This 1-character field indicates to the database manager whether it should read EXTEND input file commands. Y means that the database manager should read EXTEND input file commands, and N, the default, means it should not. You can specify Y only when STARTUP=W, R, U, or F (F is valid for DB2 Server for VSE only). (Otherwise, you receive an error message.)</p> <p data-bbox="817 504 1423 707">Specify the EXTEND input file commands in either a CMS file that has fixed-length, 80-byte records, or in SYSIPT. Before starting the application server enter a CMS FILEDEF command with the ddname ARIEXTND that identifies the CMS file. The EXTEND input file commands are described in the <i>DB2 Server for VSE & VM Diagnosis Guide and Reference</i> manual.</p> <p data-bbox="817 732 1414 783">If you are starting the application server in single user mode, the single user mode job must not use SYSIPT.</p>

Parameter	Default	Description
LOGMODE= AILNIY	Y	<p>Logging Mode</p> <p>This 1-character field indicates how the database manager is to maintain the logs:</p> <p>A The database manager is to maintain the logs and automatically archive the database when the level specified by ARCHPCT is reached.</p> <p>L The database manager is to maintain the logs and automatically archive the log or switch to the inactive log disk (if alternate logging is enabled)when the level specified by ARCHPCT is reached.</p> <p>N The database manager is not to maintain the logs for recovery. A log is still needed, however, for internal application server operations. <i>A log is always needed</i>, whether for recovery of user data or not. The physical device used for the log must always be allocated, even when LOGMODE=N is specified. You cannot specify LOGMODE=N when SYSMODE=M.</p> <p>Y The database manager is to maintain the logs. It is not always possible to switch log modes. Refer to the <i>DB2 Server for VM System Administration</i>, or <i>DB2 Server for VSE System Administration</i> manual for information about switching log modes.</p> <p>For VSE: Note that if A or L is specified, DB2 Server for VSE job control must include a TLBL statement with the file name ARIARCH for the database archive tape. If you specify LOGMODE=L, the job control must also include a TLBL statement with the file name ARILARC for the active log archive tape. If alternate logging is enabled, a TLBL statement with the file name ARILALT will be needed for the inactive log archive tape. The database manager will dynamically assign the tape units unless you specified static assign in your startup JCL. See Figure 21 on page 47 for sample JCL. When dynamic assign is chosen, your JCL must not contain an ASSGN statement.</p> <p>For VM: The SQLSTART EXEC issues a CMS FILEDEF command with the ddname ARIARCH for the database archive file (which can be directed only to tape). It also issues a CMS FILEDEF command with the ddname ARILARC for the active log archive file (which can be directed to tape or disk). If alternate logging is enabled, SQLSTART also issues a CMS FILEDEF command for the ddname ARILALT for the inactive log archive file (which can also be on tape or disk). Users can enter CMS FILEDEF and LABELDEF commands for the archive file before starting the database manager. (User FILEDEFs override the FILEDEFs that SQLSTART issues.) The ddnames must be ARIARCH for database archive files and ARILARC and ARILALT for log archive files.</p>

Parameter	Default	Description
LTIMEOUT= <i>n</i>	0	<p>Lock Timeout</p> <p>Number of seconds before a lock wait will timeout.</p> <p>This initialization parameter specifies a general lock wait timeout period for any SQL application, and especially as the way to avoid global deadlocks for DUOW applications.</p> <p>The range of the LTIMEOUT value is 0 to 99999 seconds.</p> <p>The default is 0, which means no timeout.</p>
NCSCANS= <i>n</i>	30	<p>Number of Concurrent Scans</p> <p><i>n</i> is a decimal value that determines the maximum number of concurrent scans that a given user might have outstanding. A scan maintains positioning information within the database. That is, to hold information related to a DB2 Server for VSE & VM cursor. The information is kept in a scan table. The maximum size of this scan table is 32 kilobytes. The maximum number of DB2 Server for VSE & VM cursors per active agent is 255. This value might be less depending on the amount of positioning information required for a particular cursor. The average scan for a cursor is 50 bytes. <i>n</i> can be any number from 1 to 655. For more information, see the <i>DB2 Server for VSE System Administration</i>, or <i>DB2 Server for VM System Administration</i> manual.</p>
NCUSERS= <i>n</i>	5	<p>Number of Concurrent Users</p> <p><i>n</i> is a decimal value that defines the maximum number of concurrent users that the database manager can support in multiple user mode. The actual number of <i>logged-on</i> users, however, might exceed this number.</p> <p>One of the uses of this value is to generate DB2 Server for VSE & VM task control blocks called <i>agent structures</i> (or <i>agents</i>). Normally, one user is associated with one agent structure.</p>
NDIRBUF= <i>n</i>	NPAGBUF	<p>Number of Directory Buffers</p> <p><i>n</i> is a decimal value that defines how many directory buffers are to be reserved in virtual storage. These buffers are shared by all users. Each buffer requires 512 bytes. <i>n</i> can be any number from 10 to 400000.</p>

Parameter	Default	Description
NLRBS= <i>n</i>	Calculated	<p>Number of Lock Request Blocks for the Application Server</p> <p><i>n</i> is a decimal value that defines the maximum number of lock request blocks available in multiple user mode. This value limits the total number of locks that can be requested by all concurrent users. When the application server approaches this value (comes to within 2*NCUSERS of the value), the lock escalation process occurs. The default value is NCUSERS*2 + (NCUSERS*NLRBU)/2 + 10. The minimum value is the greater of 50 or NCUSERS*2. The maximum value is 583333.</p>
NLRBU= <i>n</i>	1000	<p>Number of Lock Request Blocks per User</p> <p><i>n</i> is a decimal value that defines the maximum number of lock request blocks available to any one user in multiple user mode. When the application server reaches this value for any application, the lock escalation process occurs. (See the LOCKLMT and ESCALATE counters under the COUNTER command on page 58 for more details about lock escalation.) In general, you can assume that for every lock that an application holds, two lock request blocks are required. The minimum value is 10, and the maximum is 583,333.</p>
NPACKAGE= <i>n</i>	10	<p>Maximum Number of Active Packages in a LUW</p> <p><i>n</i> is a decimal value that determines the maximum number of active packages in an LUW. The value of <i>n</i> must be in the range from 1 to <i>m</i> where (<i>m</i> x NCUSERS) must be less than 32767. For more information see the <i>DB2 Server for VSE System Administration</i>, or <i>DB2 Server for VM System Administration</i> manual.</p>
NPACKPCT= <i>n</i>	30	<p>Packages in Cache Threshold</p> <p><i>n</i> is an integer value indicating the number of packages which will remain loaded in cache as a percentage of the total cache. <i>n</i> can be in the range of 0 to 100 percent.</p> <p>Size of package cache= NPACKAGE x NCUSERS</p> <p>Threshold = size of package cache x NPACKPCT / 100</p> <p>When the number of loaded packages exceeds the threshold, packages will be discarded at the end of the LUW until the number of loaded packages drops below the threshold level. The value of the threshold is truncated to an integer. If the threshold is 0, one package will be kept in cache. For more information see the <i>DB2 Server for VSE System Administration</i>, or <i>DB2 Server for VM System Administration</i> manual.</p>

Parameter	Default	Description
NPAGBUF= <i>n</i>	Calculated	Number of Data Page Buffers <i>n</i> is a decimal value that defines how many data page buffers are to be reserved in virtual storage. These buffers are shared by all users. Each buffer requires 4 096 bytes. The default value for <i>n</i> is (10 + (4 x NCUSERS)). <i>n</i> can be any number from 10 to 400,000.
PARMID= <i>name</i>	None	Name of Parameter File A 1- to 8-character name that defines a DB2 Server for VSE source member or a DB2 Server for VM CMS file that contains initialization parameters that are to override the default values. (The VM file type must be SQLPARM.)
PROCXAB= <i>nmn</i>	0	Number of Stored Procedures <i>nmn</i> is a decimal value specifying the number of times a stored procedure is allowed to terminate abnormally, after which a STOP PROC ACTION REJECT is performed against the procedure and all subsequent SQL CALL statements are rejected. Note that a timeout that occurs while waiting for a stored procedure server to be assigned for an SQL CALL statement is not included in this count. The default, 0, means that the first abend of a stored procedure causes SQL CALLs to that procedure to be rejected. For production systems, you should accept the default.
PROGNAME= <i>name</i>	None	Program Name A 1- to 8-character name that defines the program to be loaded and called by the application server when the application server is initialized. The name is required for single user mode only when STARTUP=W, U, F, or R. (F is valid for DB2 Server for VSE only.) It can be the name of an application, the DBS Utility program, or a DB2 Server for VSE & VM preprocessor.
PTIMEOUT= <i>nnnnn</i>	180	PSERVER Timeout This parameter serves two purposes: <ol style="list-style-type: none"> 1. The number of seconds before DB2 Server for VSE & VM ceases to wait for an SQL CALL to be assigned to a stored procedure server. If the PTIMEOUT interval expires, the SQL statement fails, and SQLCODE -913 is returned with SQLSTATE 40001. 2. The number of seconds before DB2 Server for VSE & VM ceases to wait for the START PSERVER command to complete. If the PTIMEOUT interval expires, message ARI4168I is displayed and the START PSERVER command terminates. A value of 0 means that no PTIMEOUT is in effect.

Parameter	Default	Description
SLOGCUSH= <i>n</i>	90	<p>DB2 Server for VSE & VM Log Cushion</p> <p><i>n</i> is a decimal value indicating percentage. When the log becomes <i>n</i> percent full, the database manager invokes the log overflow procedure. The log overflow procedure is described in the <i>DB2 Server for VSE System Administration</i>, and <i>DB2 Server for VM System Administration</i> manuals. <i>n</i> can be any number from 11 to 100. For optimum use of the archive function, the value for SLOGCUSH should be <i>greater than</i> the value of the ARCHPCT parameter.</p>
SOSLEVEL= <i>n</i>	10	<p>Short-on-Storage Level</p> <p><i>n</i> is a decimal value indicating percentage. For LOGMODE=N, message ARI0202I is issued when <i>n</i> percent of the storage pool pages remain free. For every other LOGMODE, when <i>n</i> percent or less of the pages in the storage pool are free, the database manager takes a checkpoint in an attempt to free shadow pages. If, after this checkpoint, there are still only <i>n</i> percent or less free pages in the storage pool, message ARI0202I is issued once. When this message occurs, the system should be shut down as soon as possible so that more disk storage can be added to the particular storage pool (with the ADD DBEXTENT function). <i>n</i> can be any number from 1 to 100.</p>

Parameter	Default	Description
STARTUP= <i>type</i>	W	<p>Type of Startup</p> <p>This 1-character field indicates how the application server is to be started:</p> <p>C Database or catalog generation (cold start). The database manager formats the entire database, including the log.</p> <p>E Add dbextents, delete dbextents, or both.</p> <p>F Restore the database from an archive tape without reformatting the database files. This can be specified only when none of the database files (BDISK, DDSK1-n) have been replaced. Type F is valid in VSE only.</p> <p>I Reorganize the indexes on the catalog tables.</p> <p>L Redefine the log data sets (COLDLOG). The LOGMODE must be set to Y.</p> <p>M Migrate a DB2 Server for VSE & VM database from a previous to a current version of the product.</p> <p>P Release empty pages.</p> <p>R Restore the database from an archive tape.</p> <p>S Add new dbspaces.</p> <p>U User restore. Use this parameter to start the database manager if you have just restored the database from a user archive.</p> <p>The storage pool will be brought up-to-date with the rest of the database if you are using the Data Restore Feature. See the <i>DB2 Server for VSE & VM Data Restore</i> for more information.</p> <p>W Normal start (warm start).</p> <p>Cold start (C) or COLDLOG (L) cancels the restart recovery procedure.</p> <p>Warm start (W), archive restore (R or F), or user restore (U) can be specified in either multiple user or single user mode. The other options can be specified only in single user mode (SYSMODE=S).</p> <p>Note: In VM, you should not normally specify STARTUP=C, L, E, I, M, or S. The database manager provides EXECs for each of these functions. The startup parameters are used as part of a process to perform each of the functions, but the parameters do not, in themselves, perform the entire function.</p> <p>For descriptions of the EXECs for values C, E, L, R, S, U and W, see the <i>DB2 Server for VM System Administration</i> manual; for values I and P see the <i>DB2 Server for VSE & VM Database Administration</i> manual, and for value M see the <i>DB2 Server for VM Program Directory</i>.</p>

Parameter	Default	Description
SYNCPNT=Y N	Y	<p>Sync point manager (SPM) support</p> <p>This 1-character field indicates whether or not a sync point manager (SPM) will be used to coordinate DRDA-2 DUOW two-phase commit and resynchronization activity.</p> <p>Y means that the server will use a sync point manager, if possible, to coordinate the two-phase commits and resynchronization activity.</p> <p>N means that the server will not use an SPM to perform two-phase commits. Also if N is specified, DB2 Server for VSE & VM is limited to multi-read, single-write distributed units of work and it can be the single-write site.</p> <p>Note: If Y is specified, but DB2 Server for VSE & VM finds that a sync point manager is <i>not</i> available, then the server will operate as if N was specified.</p>
SYSMODE=M S	M	<p>System Mode</p> <p>This 1-character field indicates whether the application server is to run in single user mode (S) or multiple user mode (M).</p>
SECALVER=Y N	N	<p>Security</p> <p>The value of this parameter is determined by the value of the SECTYPE parameter. If SECALVER=Y is specified, and the requester sent a userid but no password for authentication, the application server assumes that the user is already verified. If SECALVER=N is specified, the application server assumes that the user is not already verified and a password or PassTicket must accompany a userid for authentication.</p> <p>This parameter is valid only when the incoming connect request is via TCP/IP from any type of requester or when the incoming connect request is from a DB2 Server for VSE DRDA application requester via SNA.</p>
SECTYPE=DB2 ESM	DB2	<p>Security</p> <p>If SECTYPE=DB2 is specified, the SYSTEM.SYSUSERAUTH table is checked if the user has connect authority; the password column is not checked.</p> <p>If SECTYPE=ESM is specified, a check is performed during initialization to ensure that the external security manager is available and that it supports the RACROUTE interface.</p> <p>This parameter is valid only when the incoming connect request is via TCP/IP from any type of requester or when the incoming connect request is from a DB2 Server for VSE DRDA application requester via SNA.</p>

Parameter	Default	Description
TAPEMGR=N Y	N	<p>Tape Manager</p> <p>This 1 character field indicates whether there is a tape manager available to handle tape assigns during database and log archives. Y indicates there is a tape manager; N indicates there is no tape manager.</p> <p>If TAPEMGR=N, then the operator will be prompted to enter the virtual device address of the archive output.</p> <p>If TAPEMGR=Y, then the tape assign will automatically be handled by the tape manager and the operator will not be prompted to enter the virtual device number (cuu). In VM, log archives will automatically be directed to tape if TAPEMGR = Y.</p>
TCPPORT= <i>nnnnn</i>		<p>This identifies the TCP/IP port number as an integer in the range of 1 to 65535.</p> <p>If TCPPORT=0 is specified, no TCP/IP initialization will be attempted.</p> <p>For VSE, if TCPPORT is not specified, the DBNAME directory will determine whether TCP/IP initialization is required and what port number should be used. For VM, if TCPPORT is not specified, the ETC SERVICES file is used to determine what port number should be used.</p> <p>Note: For VSE, the same TCP/IP port number can be used among different application servers to initialize TCP/IP support. In this case, the last server using the duplicate port number will become the target server owning the port. All later incoming connect requests using this port number will be connected to this target server. To avoid misdirected connection requests, duplicate port numbers should not be used.</p>
TCPMAXRT= <i>n</i>	158	<p>Maximum number of TCP/IP support retry attempts.</p> <p><i>n</i> is the maximum number of times that automatic restart of TCP/IP support is attempted.</p> <p>The range of the TCPMAXRT value is 1 to 9999.</p> <p>The default is 158.</p> <p>Note: If the TCPRETRY parameter value is N, the value of TCPMAXRT is ignored by the database manager.</p>

Parameter	Default	Description
TCPRETRY=Y N	Y	<p>Automatic restart if TCP/IP support fails.</p> <p>This 1 character field indicates whether or not the database manager will try to automatically restart TCP/IP support if it fails.</p> <p>Y indicates the database manager will try to automatically restart TCP/IP support if it fails.</p> <p>N indicates the database manager will not try to automatically restart TCP/IP support if it fails.</p> <p>N indicates the database manager will not try to automatically restart TCP/IP support if it fails.</p> <p>Note: If Y is specified, but TCPSPORT = 0, then the database manager will change the value to N.</p>

Common Tracing Parameters

During DB2 Server for VSE initialization, the application server prompts you for the tape cuu or DISK. The DISK reply causes the application server to open a DASD SAM file for tracing. If any tracing is to be done, DB2 Server for VSE job control must include a TLBL or DLBL statement with the file name ARITRAC.

During DB2 Server for VM initialization, the SQLSTART EXEC supplies this (default) CMS FILEDEF command: FILEDEF ARITRAC TAP2 SL (BLKSIZE 4096 NOCHANGE PERM

Parameter	Default	Description
TRACDBSS= nnnnnnnnnnn	Zeros	<p>Tracing in DBSS</p> <p>This 11-digit parameter specifies the parts of the DBSS that are to be traced and the level. Where 0 is specified, tracing is not done. Where 1 is specified, tracing displays only module entry and exit points. Where 2 is specified, tracing is done in greater detail. The positional digits correspond to the following DBSS functions and subcomponents: ENTRY, EXIT, LOG, LOCK, LUW, DC, DM, STOR, SORT, INDEX, and STAT. (A description of these is in "Starting a Trace" on page 160.)</p>
TRACDRRM=nnnn	Zeros	<p>Tracing of the DRRM component</p> <p>This four-digit parameter specifies the parts of the DRRM that are to be traced and the level. Where 0 is specified, tracing is not done. Where 1 is specified, tracing displays only module entry and exit points. Where 2 is specified, tracing is done in greater detail. The positional digits correspond to the following DRRM functions and subcomponents: the first is the DDM Parser, the second is the DDM GENERATOR, the third is the DDM Data Dictionary, and the fourth is the RDIIN Manager. (A description of these is in "Starting a Trace" on page 160.)</p> <p>Note: You will only receive trace output for the work unit manager if you specified AUTO for the PROTOCOL parameter.</p>

Parameter	Default	Description
TRACDSC= <i>mm</i>	Zeros	<p>Tracing of the DSC component</p> <p>This two-digit parameter specifies the parts of the DSC component to be traced and the level. Where 0 is specified, no tracing is done. Where 1 is specified, tracing is done in limited detail. Where 2 is specified, tracing is done in greater detail. The positional digits correspond to the following DSC functions and subcomponents: AGENT and COM. (A description of these is in "Starting a Trace" on page 160.)</p>
TRACEBUF= <i>n</i>	Zero	<p>Tracing in Memory</p> <p>The parameter <i>n</i> specifies the amount of memory in kilobytes that you want to allocate to the trace buffer. The range of the TRACEBUF integer value <i>n</i> is 0 to 99999. A suggested size for the trace buffer is 100 kilobytes or more. It is recommended that you increase either the Virtual Machine memory or the size of the partition by the size of the trace buffer before you start a trace in memory. The trace buffer will only be created if you specify TRACEBUF with at least one of the startup initialization parameters TRACCONV, TRACRDS, TRACDBSS, TRACDSC, TRACDRRM, TRACSTG, or TRACWUM. Trace records are stored in wrap-around mode in a fixed size buffer. A trace buffer will not be created if the TRACEBUF default or <i>n</i>=0 is selected.</p> <p>If there are records in the buffer, they are written to the trace output file by the TRACE OFF command or by DB2 Server for VSE & VM shut down processing. For details see SQLEND and TRACE OFF operator commands on pages 11 and 165 respectively.</p> <p>Note: If there is insufficient storage available for the trace buffer, error message ARI0155E will be displayed on the operator console and the system will not be started.</p>
TRACRDS= <i>nnnnnnn</i>	Zeros	<p>Tracing in RDS</p> <p>This seven-digit parameter specifies the parts of the RDS to be traced and the level. Where 0 is specified, tracing is not done. Where 1 is specified, tracing displays only module entry and exit points. Where 2 is specified, tracing is done in greater detail. The positional digits correspond to the following RDS functions and subcomponents: EXEC, PA, OPT, AG, INT, AU, and SG. (A description of these is in "Starting a Trace" on page 160.)</p>
TRACSTG= <i>n</i>	Zero	<p>Tracing in the System and Working storage</p> <p>This parameter can be specified on the application server side only. It can be invoked during startup when the database is first initialized and with the operator TRACE command. The TRACSTG parameter has no subcomponents. Values of <i>n</i> are 0 which specifies no trace and 1 which specifies a level 1 trace.</p>

Parameter	Default	Description
TRACWUM= <i>n</i>	Zero	<p>Tracing in the Work Unit Manager</p> <p>This parameter specifies the level of tracing for the work unit manager. Where 0 is specified, tracing is not done. Where 1 is specified, tracing is done for module entries and exits. Where 2 is specified, tracing is done in greater detail.</p> <p>Note: You will only receive trace output for the work unit manager if you specified AUTO for the PROTOCOL parameter.</p>
TRACCONV= <i>n</i>	Zero	<p>Tracing of the CONV component</p> <p>This parameter <i>n</i> specifies the level of tracing for the CONV component. Where 0 is specified, tracing is not done. Where 1 is specified, tracing is done in limited detail. Where 2 is specified, tracing is done in greater detail.</p>

DB2 Server for VSE Parameters

The following table contains initialization parameters specific to the VSE environment.

Parameter	Default	Description
ACCOUNT=T D E N	N	<p>Accounting</p> <p>This 1-character field indicates whether the database manager is to generate accounting records. N indicates no accounting records are to be generated. D, E or T indicates that you want the database manager to generate accounting records. D means that you want the database manager to write the records to a sequential DASD file. E means that you want the database manager to write records to a VSAM ESDS file. T means that you want the database manager to write the records to a tape file.</p> <p>If you specify T, D, or E, your job control for starting the application server must include appropriate TLBL, DLBL, EXTENT, and ASSGN statements. The <i>DB2 Server for VSE System Administration</i> manual describes how to set up accounting files and appropriate job control.</p>
ARCHTAPE=UNL REW	REW	<p>Archive Tape</p> <p>This field controls when archive tapes are unloaded. UNL indicates that at the end of writing to each tape of a log or database archive, the tape will be unloaded from the tape drive. This occurs regardless of whether the archives are dynamically or statically assigned.</p> <p>REW indicates that only archives taken using dynamic allocation will be unloaded at end of tape. If it is the last tape of the archive, and end of tape has not been reached, the tape will remain in the drive and must be unloaded manually by the operator.</p>

Parameter	Default	Description
DBNAME= <i>server-name</i>	SQLDS	<p>Server-Name</p> <p>This optional 1- to 18-character field specifies the name of the application server to be accessed. It is used by the database manager to verify that the application requesters are accessing the intended application server when they attempt to connect to the database machine.</p> <p>If the DBNAME is not specified, the partition or system default will be assumed.</p>
DBPSWD= <i>password</i>	None	<p>Database Password</p> <p>The password is a 1- to 8-character field that provides the VSAM password for the database. If a VSAM CONTROLPW or MASTERPW password was defined for the DB2 Server for VSE data sets at database generation, the password must match it exactly. This value is not displayed with the others during initialization.</p>
DSPLYDEV=L C B	L	<p>Display Device</p> <p>This one-character field indicates where output such as the initialization parameter values and the minidump is to be routed. L indicates SYSLST. C indicates SYSLOG. B indicates both SYSLST and SYSLOG.</p>
RMTUSERS= <i>nnnnn</i>	0	<p>Maximum number of remote users accessing the application server</p> <p>This parameter lets you specify the number of remote users that can access the application server. The allowable range of users is 0 to 65535; the practical value of RMTUSERS depends on the availability of virtual storage.</p> <p>When a value has not been specified or <i>nnnnn</i> has the value zero, remote users will not be able to access the application server.</p>

Parameter	Default	Description
TCPDISPB= <i>n</i>	1	<p>TCP/IP Dispatcher Bias</p> <p>This value specifies how frequently the TCPIP agent is dispatched when there are no agents ready to run and a system wait must be done. The TCPIP agent will be dispatched every <i>n</i> times a system wait must be done. Otherwise, the system wait is performed by the DB2/VSE dispatcher.</p> <p>Valid values are 1 to 100.</p> <p>Setting the value to a high number can improve the performance of agents that are doing I/O intensive queries and are not using TCP/IP as the communications protocol. This setting will not affect the performance of CPU intensive queries as much as it will affect I/O intensive queries.</p> <p>Setting the value to a high number can degrade the performance of users that are using TCP/IP as the communications protocol.</p> <p>This value is ignored if RMTUSERS=0. It is only active if TCP/IP support is enabled.</p>

DB2 Server for VM Parameters

Refer to “Starting the Application Server in a VM Environment” on page 2 for the format of the SQLSTART EXEC syntax diagram and an explanation of the SQLSTART EXEC options.

Parameter	Default	Description
ACCOUNT=D N	N	<p>Accounting</p> <p>This 1-character field indicates whether the database manager is to generate accounting records. N indicates no accounting records are to be generated. D indicates that you want the database manager to generate records. The database manager writes (spools) the accounting records to the VM system accounting file.</p>

Parameter	Default	Description
DBMODE=L G N	Defined in VM directory	<p>Database Mode</p> <p>This 1-character field specifies the mode in which the database manager is used. This parameter is only used for VM/SP and VM/ESA systems.</p> <p>If DBMODE=L is specified, the database manager attempts to start the database as a local resource. There must be a directory entry that allows that resource to be identified as a local resource.</p> <p>If DBMODE=G is specified, the database manager attempts to start the database as a global resource. There must be a directory entry that allows that resource to be identified as a global resource.</p> <p>DBMODE=N is used during migration; see the <i>DB2 Server for VM System Administration</i> manual for information on DBMODE=N.</p> <p>If the DBMODE parameter is not specified, the database manager system defaults to the resource authorization specified in its VM directory.</p>
PROTOCOL= <i>type</i>	SQLDS	<p>Protocol Type</p> <p>Specifies the type of protocol that the application server is capable of handling. Use SQLDS if your database is only to be accessed through SQLDS protocol from DB2 Server for VM application requesters only. Specify the AUTO option if both DRDA and SQLDS protocols are to be supported.</p> <p>Note: The DRDA protocol is not available if you have not installed the optional DRDA code. For a description of how to do this, refer to the <i>DB2 Server for VM System Administration</i> manual.</p>
TCPCAN=Y N	Y	<p>Cancel TCP/IP Agent in VM Support</p> <p>This 1-character field indicates whether the Cancel TCP/IP Agent in VM Support is enabled or disabled. Y indicates that the cancel support is enabled. N indicates that the cancel support is disabled.</p> <p>There is some performance overhead if cancel support is enabled. This overhead may be unacceptable for some users.</p>

DB2 Server for VM Data Spaces Support Parameters

The following additional initialization parameters may be used when starting the VM Server if you have installed the optional VMDSS code to support VM Data Spaces.

Parameter	Default	Description
MAPPING=L P	L	Sets whether the database manager will use logical or physical mapping. To use logical mapping set MAPPING=L. For physical mapping set MAPPING=P.

Parameter	Default	Description
SAVEINTV= <i>n</i>	10	Sets the save interval for the database to <i>n</i> blocks of 4KB pages (32 pages in a block). The save interval is the number of blocks in a data space that the database manager modifies before it writes them to DASD. Valid values are integers from 1 to 999999 blocks.
SEPINTDB=Y N	Y	Sets whether the database manager will use separate unmapped internal dbspaces or not. To use unmapped internal dbspaces, set SEPINTDB=Y. Note, setting SEPINTDB=N does not turn Data Spaces Support off. Because mapped internal dbspaces are assigned to one storage pool, they will use whichever DASD I/O system is specified for that pool.
TARGETWS= <i>n</i>	32	Sets the target working storage size for the database to <i>n</i> megabytes. This limits the amount of main and expanded storage that the database machine will use. While the database machine may exceed this target, the database manager will keep you at or below it if possible. Valid values are integers from 1 to 999999 megabytes.

Appendix B. VMDSS Operator Commands

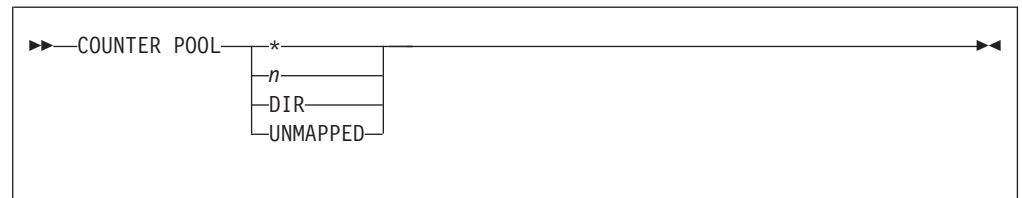
This section contains descriptions of the operator commands included with VMDSS. These commands can be used in a VM/ESA only environment.

Note: You can also use the SHOW and COUNTER commands from ISQL or REXX EXECs. You must have DB2 REXX SQL for VM/ESA (RXSQL) installed before you can use REXX with DB2 Server for VM.

COUNTER POOL

Displays the number of occurrences of key events by storage pool. Also displays counters for the directory and unmapped internal dbspaces.

Syntax



***** Displays counters for all storage pools, the directory, and unmapped internal dbspaces.

n Displays counters for storage pool number *n*. Valid values are integers from 1 to 999.

DIR Displays counters for the directory.

UNMAPPED

Displays counters for unmapped internal dbspaces.

Description

Storage Pool Counters: The COUNTER POOL command displays two sets of counters. One set is for storage pools using Data Spaces Support, a directory that uses Data Spaces Support, and unmapped internal dbspaces. The other set is for pools using the standard DASD I/O system and a directory using the standard system.

The counters for the standard I/O system are:

LBUFLOOK

The number of times the database manager has looked into the local data page buffers.

PGREAD

The number of pages moved from DASD to the local buffers.

PGWRITE

The number of pages moved from the local buffers to DASD.

IUCVBIO

The number of IUCV *BLOCKIO requests submitted by the database manager to the operating system. Note that, because the database manager often uses one IUCV *BLOCKIO request to write several pages at one time, IUCVBIO is usually lower than PGREAD+PGWRITE.

The counters for Data Spaces Support are:

LBUFLOOK

The number of times the database manager has looked into the local data page buffers.

DSREAD

The number of pages moved from the VM data spaces to the local buffers.

DSWRITE

The number of pages moved from the local buffers to the VM data spaces.

DSFAULT

The number of pages, requested from data spaces, that the operating system did not find in main or expanded storage. These page faults cause the operating system to read the pages from DASD in to the data spaces in main storage.

Example

This example shows a database where the directory and the first storage pool are using Data Spaces Support, the second pool is using the standard DASD I/O system, and the internal dbspaces are mapped. Because the internal dbspaces are included in storage pool 1, any activity recorded for them will be included with the totals for pool number 1.

```

counter pool *
Counter values at  DATE='05-24-91'  TIME='10:13:49'

Directory:  Data Spaces
Pages looked at in the buffer  LBUFLOOK:  90
Pages moved from DS to buffer  DSREAD   :  57
Pages moved from buffer to DS  DSWRITE  :  28
DS page fault notifications    DSFAULT  :  5

Pool No. 1:  Data Spaces
Pages looked at in the buffer  LBUFLOOK: 1005
Pages moved from DS to buffer  DSREAD   :  77
Pages moved from buffer to DS  DSWRITE  :  0
DS page fault notifications    DSFAULT  : 18

Pool No. 2:  *BLOCKIO
Pages looked at in the buffer  LBUFLOOK: 237
Page reads                     PGREAD   :  42
Page writes                     PGWRITE  : 12
IUCV *BLOCKIO I/O requests    IUCVBIO  :  9

Total count for storage pools using data spaces:
Pages looked at in the buffer  LBUFLOOK: 1005
Pages moved from DS to buffer  DSREAD   :  77
Pages moved from buffer to DS  DSWRITE  :  0
DS page fault notifications    DSFAULT  : 18

Total count for storage pools using *BLOCKIO:
Pages looked at in the buffer  LBUFLOOK: 237
Page reads                     PGREAD   :  42
Page writes                     PGWRITE  : 12
IUCV *BLOCKIO I/O requests    IUCVBIO  :  9
ARI0065I Operator command processing is complete.

```

This example shows a database where the directory is using the standard DASD I/O system.

```

counter pool dir
Counter values at  DATE='05-23-91'  TIME='15:41:07'

Directory:  *BLOCKIO
Pages looked at in the buffer  LBUFLOOK: 410
Page reads                     PGREAD   :  52
Page writes                     PGWRITE  : 11
IUCV *BLOCKIO I/O requests    IUCVBIO  :  9
ARI0065I Operator command processing is complete.

```

This example shows a database using unmapped internal dbspaces. Any activity recorded for these dbspaces will only appear under the heading Unmapped Internal Dbspaces. It will not be included in the totals for the storage pool that contains the internal dbspaces.

```

counter pool unmapped
Counter values at  DATE='07-29-91'  TIME='17:12:37'

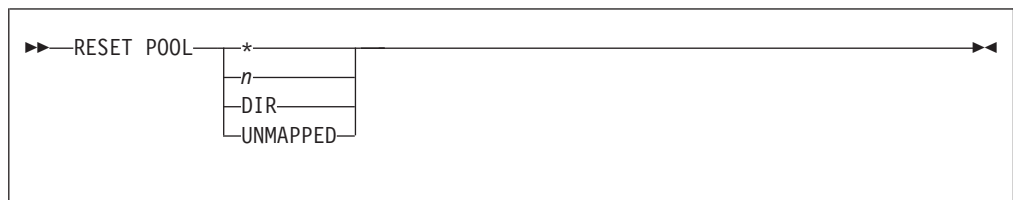
Unmapped Internal Dbspaces:  Data Spaces
Pages looked at in the buffer  LBUFLOOK:  515
Pages moved from DS to buffer  DSREAD   :  21
Pages moved from buffer to DS  DSWRITE  :   2
DS page fault notifications    DSFAULT  :   0
ARI0065I Operator command processing is complete.

```

RESET POOL

Resets storage pool counters to zero. Also resets the counters for the directory and unmapped internal dbspaces to zero.

Syntax



***** Resets the counters for all storage pools, the directory, and unmapped internal dbspaces to zero.

n Resets the counters for storage pool number *n* to zero. Valid values are integers from 1 to 999.

DIR Resets the counters for the directory to zero.

UNMAPPED

Resets counters for unmapped internal dbspaces.

Description

This operator command resets storage pool and directory counters to zero. You can use this command to develop a picture of database performance over time. At regular intervals, display the storage pool and directory counters with the COUNTER POOL command; then reset the counters with RESET POOL. The next time you display the counters, they will display the number of key events that occurred between the time when you issued the RESET POOL command and the time when you issued COUNTER POOL.

Example

```
counter pool 1
Counter values at  DATE='05-24-91'  TIME='10:11:07'

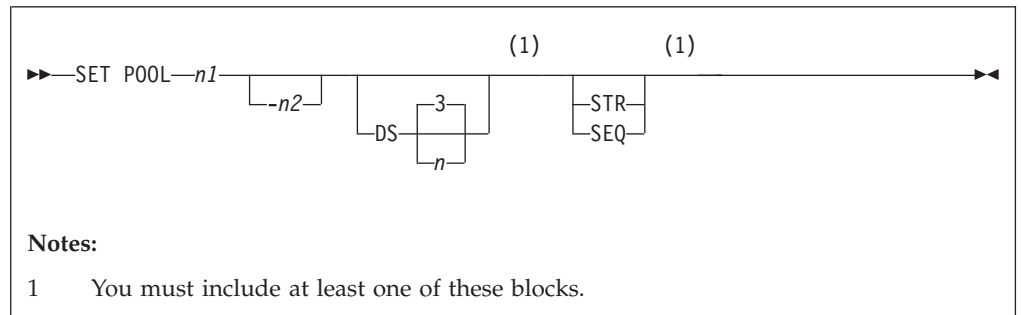
Pool No. 1: Data Spaces
Pages looked at in the buffer  LBUFLOOK: 1005
Pages moved from DS to buffer  DSREAD  : 77
Pages moved from buffer to DS  DSWRITE : 0
DS page fault notifications    DSFAULT : 18
ARI0065I Operator command processing is complete.
reset pool 1
Counters reset at  DATE='05-24-91'  TIME='10:11:15'
ARI0065I Operator command processing is complete.
counter pool 1
Counter values at  DATE='05-24-91'  TIME='10:11:22'

Pool No. 1: Data Spaces
Pages looked at in the buffer  LBUFLOOK: 0
Pages moved from DS to buffer  DSREAD  : 0
Pages moved from buffer to DS  DSWRITE : 0
DS page fault notifications    DSFAULT : 0
ARI0065I Operator command processing is complete.
```

SET POOL

Changes storage pool specifications.

Syntax



- n1** Specifies that you want to change the specifications for storage pool *n1*. Valid values are integers from 1 to 999.
- n2** Specifies a range of storage pools from *n1* to *n2*. Valid values are integers from 1 to 999. *n2* must be greater than or equal to *n1*.
- DS** Sets the working storage residency priority of the storage pools you specify to *n*. Valid values are integers from 1 to 5. The default is 3. If you do not include **DS** the database manager will use the current storage pool setting (unless the pools you specified are not using data spaces).

Table 3 on page 204 summarizes the five storage residence priorities. An R indicates that the database manager releases a page from main and expanded storage after it has been moved to a local buffer.

Table 3. Storage Residence Priorities

Page Type	Current Working Storage Size	Working Storage Residence Priority				
		1	2	3	4	5
Data	≤ target	R	R			
	> target	R	R	R	R	
Index	≤ target	R				
	> target	R	R	R		

STR Turns striping on for the storage pools you specify.

SEQ Turns striping off for the storage pools you specify. The database manager will allocate pages sequentially on DASD.

Note: If you do not include **SEQ** or **STR**, the database manager will use the current storage pool setting.

Description

This command lets you dynamically change some of the current storage pool specifications. Note that since you cannot use this command to dynamically change whether a storage pool uses Data Spaces Support, the storage pool you specify must already be using data spaces before you can use the **DS** parameter.

Example

The following example shows a summary of storage pools before and after turning striping on for storage pool 2:

```

show pool summary
POOL      TOTAL  NO. OF    NO. OF    NO. OF    %  NO. OF
NO.      PAGES PAGES USED FREE PAGES RESV PAGES USED EXTENTS SOS I/O
  1      1710   122   1588     20    7    2    D2S
  2      7467    2   7465     20    0    1    BK
FREE 1674279
ARI0065I Operator command processing is complete.
set pool 2 str
ARI0065I Operator command processing is complete.
show pool summary
POOL      TOTAL  NO. OF    NO. OF    NO. OF    %  NO. OF
NO.      PAGES PAGES USED FREE PAGES RESV PAGES USED EXTENTS SOS I/O
  1      1710   122   1588     20    7    2    D2S
  2      7467    2   7465     20    0    1    BKS
FREE 1674279
ARI0065I Operator command processing is complete.

```

This example shows storage pool 1 before and after turning striping off and setting its working storage residence priority to 2:

```
show pool 1
```

```
POOL NO. 1:      NUMBER OF EXTENTS = 2  DS3 STR
```

EXTENT NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED
1	855	74	781		8
2	855	47	808		5
TOTAL	1710	121	1589	20	7

```
ARI0065I Operator command processing is complete.
```

```
set pool 1 ds2 seq
```

```
ARI0065I Operator command processing is complete.
```

```
show pool 1
```

```
POOL NO. 1:      NUMBER OF EXTENTS = 2  DS2 SEQ
```

EXTENT NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED
1	855	74	781		8
2	855	47	808		5
TOTAL	1710	121	1589	20	7

```
ARI0065I Operator command processing is complete.
```

SET SAVEINTV

Sets the number of modified pages the database manager allows in data spaces before it writes them to DASD.

Syntax

```
▶▶—SET SAVEINTV—n—◀◀
```

- n* Sets the save interval for the database to *n* blocks of 4KB pages (32 pages in a block). When the number of modified blocks of pages in a data space exceeds this parameter, the database manager directs the operating system to save all the pages in that data space that were modified by the database manager to DASD. Valid values are integers from 1 to 999,999 blocks. The default is 10 blocks.

Description

This command lets you dynamically change the VMDSS save interval. It also displays your previous save interval and your new one.

Example

```
set saveintv 12
Previous SAVEINTV = 10
SAVEINTV = 12
ARI0065I Operator command processing is complete.
```

Note: This command is not allowed when no storage pools are using data spaces. For example, when the storage pool specification file specifies BLK for all

storage pools. This is true even if the directory is in a data space, or if unmapped data spaces are used for internal dbspaces (if SEPINTDB=Y is used), or both.

SET TARGETWS

Sets the target working storage size of the database.

Syntax

```
▶▶—SET TARGETWS—n—◀◀
```

- n* Sets the target working storage size for the database to *n* megabytes. This limits the amount of main and expanded storage that the database machine will use. While the database machine may exceed this target, the database manager will to keep you at or below it if possible. Valid values are integers from 1 to 999,999 megabytes. The default is 32 megabytes.

Description

This command lets you dynamically change the target working storage size. It also displays the following information about the database's:

- Previous target working storage size
- Current target working storage size
- Current working storage size
- How much of the current working storage is in main storage and how much is in expanded storage.

Example

```
set targetws 30
Previous TARGETWS = 32 MB
TARGETWS = 30 MB
Current Working Storage = 5 MB
                        (5 MAIN + 0 EXPANDED)
ARI0065I Operator command processing is complete.
```

SHOW ACTIVE

Displays the status of active database users.

Syntax

```
▶▶—SHOW ACTIVE—◀◀
```

Description

This command shows you whether a user agent is waiting for a data space page fault or the checkpoint agent is waiting for an SLD block. If a user agent is waiting for a data space page fault, you will see:

```
Agent ... is in DS page fault Wait
```

The checkpoint agent is waiting for an SLD block, you will see:

```
Checkpoint agent is in DS SLD block Wait
```

Example

```
show active
Status of SQL/DS agents:
Checkpoint agent is not active
User Agent: 1 User-ID: SMITH is R/W APPL 7B4
Agent is processing and is in I/O Wait
User Agent: 2 User-ID: MICHAEL is R/O SUBS 7B9
Agent is not processing and is in communication Wait
User Agent: 3 User-ID: JESSICA is R/O APPL 5A4
Agent is processing and is in DS page fault Wait
User Agent: 4 User-ID: TESTUSER is R/W APPL 7BB
Agent is processing and is in DS page fault Wait
4 agent(s) not connected to an APPL or SUBSYS
ARI0065I Operator command processing complete
```

```
show active
Status of SQL/DS agents:
Checkpoint agent is in DS SLD block Wait
User Agent: 1 User-ID: PETER is NEW APPL 106
Agent is waiting for log archive checkpoint. Wait
User Agent: 2 User-ID: GENE is NEW SUBS 107
Agent is waiting for log archive checkpoint.
3 agent(s) not connected to an APPL or SUBSYS
ARI0065I Operator command processing complete
```

SHOW BUFFERS

Displays information about the buffer pool size and usage.

Syntax

```
▶—SHOW BUFFERS—◀
```

Description

This command shows you whether a specific local buffer or directory buffer contains a page that comes from a data space. If a buffer contains a data space page, the **FLAGS** column will contain a hex value where the X'10' bit (00010000) is turned on.

To determine whether a particular page is from a data space, either check if the most significant digit of the flag is an odd number, or perform a logical AND between the **FLAGS** byte and X'10'. If the digit is odd, or if the result of the calculation is X'10', the page is from a data space.

For example, a **FLAGS** byte of X'D0':

```
In Hex: 'D0' & '10' = '10' or
```

```
In Binary: 11010000 & 00010000 = 00010000
```

Also the first digit of 'D0' ('D' or 13) is an odd number. This page is from a data space.

The **FLAGS** byte contains four flag bits in total:

- '80' **Modified Page.** The buffer page has been modified.
- '40' **In Transit** The buffer page is being copied to or from DASD or a data space.
- '20' **New Page** The buffer page is not from DASD or a data space; the buffer contains a new page.
- '10' **Data Space Page** The buffer page is from a data space.

The **FLAGS** byte does not use the four low order bits. They will always be zero.

Example

Note that the hex values under the **FLAGS** column are cumulative. For example, the directory buffer at address 0047A000 contains a modified (identified by X'80') data space page (identified by X'10').

```

show buffers
Only used buffers are displayed.
DBSPACE  REC      ADDR      FLAGS  FIX CNT
Page Buffers
  1      0006FC  0045C000  10     0
32001    000093  0045D000  00     1
  1      000080  0045E000  10     0
  1      00009D  0045F000  10     0
  :
  1      00009C  00478000  10     0
  1      00009B  00479000  10     0
Directory Buffers
  1      000001  0047A000  90     0
  1      000002  0047A200  00     0
  1      00000D  0047A400  00     0
  :
  6      000005  0047C800  00     0
 31      000001  0047CA00  00     0
ARI0065I Operator command processing is complete.

```

SHOW LOCK ACTIVE

Displays locking information for active database users.

Syntax

```

▶—SHOW LOCK ACTIVE—▶

```

Description

This command shows you whether a user agent is waiting for a data space page fault. If it is waiting for a data space page fault, you will see DSPF in the WAIT STATE column.

Example

```
show lock active
AGENT  USER      WAIT  TOTAL  LONG  WANTLOCK  WANTLOCK
1      JESSICA  I/O   140    100
2      LEIGH    LOCK  226    220  DBSP      12
3      MARY     COMM  97     49
4      MIKE    LOCK  108    101  DBSP      12
5      BOB     LOCK  107    98   PAGE      14
6      GENE    DSPF  88     44
7      EDWIN   DSPF  101    28
ARI0065I Operator command processing complete
```

SHOW LOCK GRAPH

Displays information about the lock dependencies between active database users.

Syntax

```
▶—SHOW LOCK GRAPH—◀
```

Description

This command shows you whether a user agent is waiting for a data space page fault. If it is waiting for a data space page fault, you will see DSPF in the WAIT STAT column.

Example

```
show lock graph
LOCK   LOCK      WAIT  LOCK  DBSP  LOCKK  REQ  REQ
REQUESTER  HOLDER  STAT  TYPE  NUMBR  QUALFIER  STATE  MODE  DUR
2  KEVIN 4  MARK  DSPF  DBSP  15           G WAIT S  LONG
4  MARK 5  MIKE  LOCK  DBSP  16           G WAIT X  LONG
5  MIKE 6  LAURA LOCK  PAGE  21    88         G WAIT IS  LONG
6  LAURA 1  STEVE COMM  DBSP  1           G WAIT IX  LONG
6  LAURA 3  GRANT DSPF  DBSP  1           G WAIT IX  LONG
ARI0065I Operator command processing complete
```

SHOW POOL

Displays physical storage information about database storage pools.

Syntax

```
▶—SHOW POOL—  
┌──ALL──┐  
├──SUMMARY──┤  
├──DELETED──┤  
└──n──┘
```

ALL Displays detailed information, grouped by dbextent, about each storage pool in the database.

SUMMARY

Displays summary information about each storage pool in the database.

DELETED

Display information about deleted dbextents

n Displays detailed information, grouped by dbextent, about storage pool number *n*.

Description

This command is updated to let you display the following for storage pool information:

- Whether it is using data spaces
- Which working storage residency priority it is using
- Whether it is using striping.

SHOW POOL SUMMARY: A column entitled "I/O" is added to the SHOW POOL SUMMARY display. Under its heading is a two-or three-letter code that indicates the current storage pool specifications. Possible specification code combinations are:

BK The storage pool is using the standard I/O system, without striping.

BKS The storage pool is using the standard I/O system, with striping.

D*n* The storage pool is using Data Spaces Support, without striping. *n* is the current storage residency priority.

D*n*S The storage pool is using Data Spaces Support, with striping. *n* is the current storage residency priority.

SHOW POOL and SHOW POOL ALL: Two new codes are added to the SHOW POOL and the SHOW POOL ALL displays beside the number of extents. Possible specification code combinations are:

BLK SEQ

The storage pool is using the standard I/O system, without striping.

BLK STR

The storage pool is using the standard I/O system, with striping.

DS*n* SEQ

The storage pool is using Data Spaces Support, without striping. *n* is the current storage residency priority.

DS*n* STR

The storage pool is using Data Spaces Support, with striping. *n* is the current storage residency priority.

Example

show pool 1

POOL NO. 1: NUMBER OF EXTENTS = 2 DS2 STR

EXTENT NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED
1	855	77	778		9
2	855	45	810		5
TOTAL	1710	122	1588	20	7

ARI0065I Operator command processing is complete.

show pool all

POOL NO. 1: NUMBER OF EXTENTS = 2 DS2 STR

EXTENT NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED
1	855	77	778		9
2	855	45	810		5
TOTAL	1710	122	1588	20	7

POOL NO. 2: NUMBER OF EXTENTS = 1 BLK STR

EXTENT NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED
3	7467	2	7465		0
TOTAL	7467	2	7465	20	0

FREE AREAS: NUMBER OF DELETED EXTENTS = 0

EXTENT NO.	TOTAL PAGES
END	1674279
TOTAL	1674279

Maximum number of DBEXTENTS = 6

ARI0065I Operator command processing is complete.

show pool summary

POOL NO.	TOTAL PAGES	NO. OF PAGES USED	NO. OF FREE PAGES	NO. OF RESV PAGES	% PAGES USED	NO. OF EXTENTS	SOS	I/O
1	1710	122	1588	20	7	2		<u>D2S</u>
2	7467	2	7465	20	0	1		<u>BKS</u>
FREE	1674279							

ARI0065I Operator command processing is complete.

SHOW SAVEINTV

Displays the current number of modified blocks of pages the database manager allows in data spaces before it writes them to DASD.

Syntax

```
▶—SHOW SAVEINTV—◀
```

Description

This command displays the database's current save interval. The save interval is measured in blocks of 4KB pages (32 pages in a block). The save interval is the upper limit on the number of modified blocks allowed in a data space at any one time. When the number of modified blocks in any one data space exceeds this limit, the database manager writes all the modified blocks in that data space to DASD. The default is 10 blocks (or 320 pages).

Note: For a block to be "modified" the database manager only needs to change one page in that block. The block will remain "modified" until it is written to DASD.

Note: This command is not allowed when no storage pools are using data spaces. For example, when the storage pool specification file specifies BLK for all storage pools. This is true even if the directory is in a data space, or if unmapped data spaces are used for internal dbspaces (if SEPINTDB=Y is used), or both.

Example

```
show saveintv
SAVEINTV = 10
ARI0065I Operator command processing is complete.
```

SHOW TARGETWS

Displays the current target working storage size of the database.

Syntax

```
▶—SHOW TARGETWS—◀
```

Description

This command displays the following information about the database's:

- Its current target working storage size
- Its current working storage size
- How much of the current working storage is in main storage and how much is in expanded storage.

The target working storage size and the working storage size are measured in megabytes. The former is the amount of main and expanded storage that the database manager will try to use for the database. The default is 32 megabytes.

Example

```
show targetws
TARGETWS = 32 MB
Current Working Storage = 5 MB
                        (5 MAIN + 0 EXPANDED)
ARI0065I Operator command processing is complete.
```

Internal Counters

VMDSS contains a set of internal counters that are primarily intended for IBM's use. However, if you have a good understanding of the CP Macros that control data spaces within VM/ESA, you may find the following counters useful.

ESASEG

The number of data space segments mapped. Data spaces are divided into 256-page chunks called segments. A segment is mapped the first time the database manager references a dbspace page that lies in that segment (the first time it references a dbspace, the database manager decides where in the data space the page will be). Mapping a segment consists of defining a mapping between every dbspace page in that segment and its place on disk. Empty pages are not mapped (because there is no disk page allocated for an empty page). Even if all the dbspace pages in a data space segment are empty, the database manager still marks that segment as mapped and increments ESASEG.

MAPSEPG

The number of calls to CP's MAPMDISK DEFINE macro to map the above ESASEG segments. It may take 0 (all pages empty) or many DEFINES per segment, because one DEFINE can only work on a contiguous set of data space pages and database manager must skip empty pages.

REQMAPP

The number of requests to remap a page. When the database manager gets a new disk page for a dbspace page (either when database manager shadows it or starts using a previously empty dbspace page), it must redefine the data space mapping for that page to point to the new disk location.

MAPP

The number of MAPMDISK DEFINE macro calls done to satisfy the REQMAPP requests above. The database manager tries to delay the requests as long as possible in order to group as many as it can into one DEFINE. In the worst case, there will be 1 DEFINE for every request. In the best case, 1 for every 256 requests.

SAVESLD

The number of calls to CP's MAPMDISK SAVE macro to save the contents of a page in the Data Space on DASD.

SAVEMXRN

The sum of "maximum runs". Before issuing a MAPMDISK SAVE macro for one or more blocks of 32 pages, a maximum run is computed. It is the largest number of contiguous blocks of 32 pages that are being saved. SAVEMXRN is an indicator of whether the blocks of pages being saved are consecutively modified (during a data load for example) and determines the span length to use for the REFPAGE macro following completion of MAPMDISK SAVE macro (refer to SCBLKREF and SCSPNREF).

SAVEBLK

The number of 32 page blocks for which CP's MAPMDISK SAVE macro was issued. The MAPMDISK SAVE macro is issued for modified blocks of pages at checkpoints and when the number of modified blocks reaches the SAVEINTV threshold.

WAITSLD

The number of times the database manager was forced to wait for a MAPMDISK SAVE macro to save modified pages on DASD. To ensure that

all modified data is safely stored on DASD, the checkpoint processing cannot finish until all outstanding MAPMDISK SAVE requests are complete.

REFBLOCK

The number of times a block form of CP's REFPAGE macro was issued telling CP the reference pattern for a group of pages.

REFBPAGE

The number of pages for which a block form of REFPAGE macro was issued.

REFBSPAN

The sum of spans used in block form of REFPAGE macro.

REFLST

The number of times a list form of CP's REFPAGE macro was issued.

REFLPAGE

The number of pages for which a list form of REFPAGE macro was issued.

DIAG10

The number of DIAGNOSE X'10' instructions processed that release the main or expanded storage back to CP.

DIAG10PG

The number of data space pages that have been released using the DIAGNOSE X'10' instruction.

SCDIAG10

The number of blocks of 32 pages for which storage has been released following the completion of MAPMDISK SAVE request.

SCREF

The number of blocks of 32 pages for which a reference pattern was established following the completion of MAPMDISK SAVE request.

SCBLKREF

The number of times a block form of the REFPAGE macro was issued following the completion of MAPMDISK SAVE request.

SCSPNREF

The sum of spans used in the block form of the REFPAGE macro following the completion of the MAPMDISK SAVE request. The size of each span is based on the largest number of consecutive blocks of 32 pages that were (SAVEMXRN counter). As the number of consecutively modified blocks increases (during a data load for example) the size of the span given to CP's REFPAGE macro also increases.

SAVEGNRL

The number of save requests made by general agents when the SAVEINTV has been reached for a data space.

SAVECHK0

The number of save requests made by the checkpoint agent to flush the page buffers prior to starting checkpoint processing.

SAVECHK1

The number of save requests made by the checkpoint agent during checkpoint processing.

To display the counters, type the following at the operator console:

```
COUNTER INTERNAL *
```

For example:

```
counter internal *
Counter values at  DATE='07-23-91'  TIME='08:57:51'
ESASEG  429
MAPSEGPG 432
REQMAPPG 15
MAPPG    13
SAVESLD  1
SAVEMXRN 1
SAVEBLK  1
WAITSLD  0
REFBLOCK 860
REFBPAGE 163912
REFBSPAN 68485
REFLST   1580
REFLPAGE 101120
DIAG10   1580
DIAG10PG 101120
SCDIAG10 0
SCREF    0
SCBLKREF 0
SCSPNREF 0
SAVEGNRL 0
SAVECHK0 0
SAVECHK1 0
ARI0065I Operator command processing is complete.
```

To reset the counters to zero, type the following:

```
RESET INTERNAL *
```

For example:

```
reset internal *
Counters reset at  DATE='05-23-91'  TIME='09:26:15'.
counter internal *
Counter values at  DATE='07-23-91'  TIME='09:26:51'
ESASEG  0
MAPSEGPG 0
REQMAPPG 0
MAPPG    0
SAVESLD  0
SAVEMXRN 0
SAVEBLK  0
WAITSLD  0
REFBLOCK 0
REFBPAGE 0
REFBSPAN 0
REFLST   0
REFLPAGE 0
DIAG10   0
DIAG10PG 0
SCDIAG10 0
SCREF    0
SCBLKREF 0
SCSPNREF 0
SAVEGNRL 0
SAVECHK0 0
SAVECHK1 0
ARI0065I Operator command processing is complete.
```

Notices

IBM may not offer the products, services, or features discussed in this document in all 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 10594-1785
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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates 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.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries:

AIX	CICS
CICS/VSE	DATABASE 2
DataPropagator	DB2
Distributed Relational Database Architecture	
DRDA	IBM
IMS	OS/2
OS/400	OS/390
SQL/DS	System/370
System/390	VM/ESA
VSE/ESA	VTAM
QMF	

Lotus and Lotus Notes are trademarks of Lotus Development Corporation 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.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

This bibliography lists publications that are referenced in this manual or that may be helpful.

DB2 Server for VM Publications

- *DB2 Server for VSE & VM Application Programming*, SC09-2889
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2983
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*, LC09-2907
- *DB2 Server for VSE & VM Overview*, GC09-2995
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2990
- *DB2 Server for VSE & VM Master Index and Glossary*, SC09-2890
- *DB2 Server for VM Messages and Codes*, GC09-2984
- *DB2 Server for VSE & VM Operation*, SC09-2986
- *DB2 Server for VSE & VM Quick Reference*, SC09-2988
- *DB2 Server for VM System Administration*, SC09-2980
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2987
- *DB2 Server for VSE & VM SQL Reference*, SC09-2989

DB2 Server for VSE Publications

- *DB2 Server for VSE & VM Application Programming*, SC09-2889
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2983
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*, LC09-2907
- *DB2 Server for VSE & VM Overview*, GC09-2995
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2990
- *DB2 Server for VSE & VM Master Index and Glossary*, SC09-2890
- *DB2 Server for VSE Messages and Codes*, GC09-2985
- *DB2 Server for VSE & VM Operation*, SC09-2986

- *DB2 Server for VSE System Administration*, SC09-2981
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2987
- *DB2 Server for VSE & VM SQL Reference*, SC09-2989

Related Publications

- *DB2 Server for VSE & VM Data Restore*, SC09-2991
- *DRDA: Every Manager's Guide*, GC26-3195
- *IBM SQL Reference, Version 2, Volume 1*, SC26-8416
- *IBM SQL Reference*, SC26-8415

VM/ESA Publications

- *VM/ESA: General Information*, GC24-5745
- *VM/ESA: VMSES/E Introduction and Reference*, GC24-5837
- *VM/ESA: Installation Guide*, GC24-5836
- *VM/ESA: Service Guide*, GC24-5838
- *VM/ESA: Planning and Administration*, SC24-5750
- *VM/ESA: CMS File Pool Planning, Administration, and Operation*, SC24-5751
- *VM/ESA: REXX/EXEC Migration Tool for VM/ESA*, GC24-5752
- *VM/ESA: Conversion Guide and Notebook*, GC24-5839
- *VM/ESA: Running Guest Operating Systems*, SC24-5755
- *VM/ESA: Connectivity Planning, Administration, and Operation*, SC24-5756
- *VM/ESA: Group Control System*, SC24-5757
- *VM/ESA: System Operation*, SC24-5758
- *VM/ESA: Virtual Machine Operation*, SC24-5759
- *VM/ESA: CP Programming Services*, SC24-5760
- *VM/ESA: CMS Application Development Guide*, SC24-5761
- *VM/ESA: CMS Application Development Reference*, SC24-5762
- *VM/ESA: CMS Application Development Guide for Assembler*, SC24-5763
- *VM/ESA: CMS Application Development Reference for Assembler*, SC24-5764

- VM/ESA: CMS Application Multitasking, SC24-5766
- VM/ESA: CP Command and Utility Reference, SC24-5773
- VM/ESA: CMS Primer, SC24-5458
- VM/ESA: CMS User's Guide, SC24-5775
- VM/ESA: CMS Command Reference, SC24-5776
- VM/ESA: CMS Pipelines User's Guide, SC24-5777
- VM/ESA: CMS Pipelines Reference, SC24-5778
- VM/ESA: XEDIT User's Guide, SC24-5779
- VM/ESA: XEDIT Command and Macro Reference, SC24-5780
- VM/ESA: Quick Reference, SX24-5290
- VM/ESA: Performance, SC24-5782
- VM/ESA: Dump Viewing Facility, GC24-5853
- VM/ESA: System Messages and Codes, GC24-5841
- VM/ESA: Diagnosis Guide, GC24-5854
- VM/ESA: CP Diagnosis Reference, SC24-5855
- VM/ESA: CP Diagnosis Reference Summary, SX24-5292
- VM/ESA: CMS Diagnosis Reference, SC24-5857
- CP and CMS control block information is not provided in book form. This information is available on the IBM VM/ESA operating system home page (<http://www.ibm.com/s390/vm>).
- IBM VM/ESA: CP Exit Customization, SC24-5672
- VM/ESA REXX/VM User's Guide, SC24-5465
- VM/ESA REXX/VM Reference, SC24-5770

C for VM/ESA Publications

- IBM C for VM/ESA Diagnosis Guide, SC09-2149
- IBM C for VM/ESA Language Reference, SC09-2153
- IBM C for VM/ESA Compiler and Run-Time Migration Guide, SC09-2147
- IBM C for VM/ESA Programming Guide, SC09-2151
- IBM C for VM/ESA User's Guide, SC09-2152

Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA) Publications

- IBM VSE/ESA Administration, SC33-6505
- IBM VSE/ESA Diagnosis Tools, SC33-6514
- IBM VSE/ESA General Information, GC33-6501
- IBM VSE/ESA Guide for Solving Problems, SC33-6510

- IBM VSE/ESA Guide to System Functions, SC33-6511
- IBM VSE/ESA Installation, SC33-6504
- IBM VSE/ESA Messages & Codes, SC33-6507
- IBM VSE/ESA Networking Support, SC33-6508
- IBM VSE/ESA Operation, SC33-6506
- IBM VSE/ESA Planning, SC33-6503
- IBM VSE/ESA System Control Statements, SC33-6513
- IBM VSE/ESA System Macros User's Guide, SC33-6515
- IBM VSE/ESA System Macros Reference, SC33-6516
- IBM VSE/ESA System Utilities, SC33-6517
- IBM VSE/ESA Unattended Node Support, SC33-6512
- IBM VSE/ESA Using IBM Workstations, SC33-6509

CICS/VSE Publications

- CICS/VSE Application Programming Reference, SC33-0713
- CICS/VSE Application Programming Guide, SC33-0712
- CICS Application Programming Primer (VS COBOL II), SC33-0674
- CICS/VSE CICS-Supplied Transactions, SC33-0710
- CICS/VSE Customization Guide, SC33-0707
- CICS/VSE Facilities and Planning Guide, SC33-0718
- CICS/VSE Intercommunication Guide, SC33-0701
- CICS/VSE Performance Guide, SC33-0703
- CICS/VSE Problem Determination Guide, SC33-0716
- CICS/VSE Recovery and Restart Guide, SC33-0702
- CICS/VSE Release Guide, GC33-1645
- CICS/VSE Report Controller User's Guide, SC33-0705
- CICS Transaction Server for VSE/ESA V1R1.0 Resource Definition Guide, SC33-0709
- CICS/VSE Resource Definition (Online), SC33-0708
- CICS/VSE System Definition and Operations Guide, SC33-0706
- CICS/VSE System Programming Reference, SC33-0711
- CICS/VSE User's Handbook, SX33-6079
- CICS/VSE XRF Guide, SC33-0704

CICS/ESA Publications

- *CICS/ESA General Information*, GC33-0803

VSE/Virtual Storage Access Method (VSE/VSAM) Publications

- *VSE/VSAM Commands and Macros*, SC33-6532
- *VSE/VSAM Introduction*, GC33-6531
- *VSE/VSAM Messages and Codes*, SC24-5146
- *VSE/VSAM Programmer's Reference*, SC33-6535

VSE/Interactive Computing and Control Facility (VSE/ICCF) Publications

- *VSE/ICCF Administration and Operation*, SC33-6562
- *VSE/ICCF Primer*, SC33-6561
- *VSE/ICCF User's Guide*, SC33-6563

VSE/POWER Publications

- *VSE/POWER Administration and Operation*, SC33-6571
- *VSE/POWER Application Programming*, SC33-6574
- *VSE/POWER Networking*, SC33-6573
- *VSE/POWER Remote Job Entry*, SC33-6572

Distributed Relational Database Architecture (DRDA) Library

- *Application Programming Guide*, SC26-4773
- *Architecture Reference*, SC26-4651
- *Connectivity Guide*, SC26-4783
- *DRDA: Every Manager's Guide*, GC26-3195
- *Planning for Distributed Relational Database*, SC26-4650
- *Problem Determination Guide*, SC26-4782

C/370 for VSE Publications

- *IBM C/370 General Information*, GC09-1386
- *IBM C/370 Programming Guide for VSE*, SC09-1399
- *IBM C/370 Installation and Customization Guide for VSE*, GC09-1417
- *IBM C/370 Reference Summary for VSE*, SX09-1246
- *IBM C/370 Diagnosis Guide and Reference for VSE*, LY09-1805

VSE/REXX Publication

- *VSE/REXX Reference*, SC33-6642

Other Distributed Data Publications

- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 4*, SC21-9526
- *IBM Distributed Data Management (DDM) Architecture, Implementation Programmer's Guide*, SC21-9529
- *VM/Directory Maintenance Licensed Program Specification*, GC20-1836
- *IBM Distributed Relational Database Architecture Reference*, SC26-4651
- *IBM Systems Network Architecture, Format and Protocol Reference*, SC30-3112
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *Reference Manual: Architecture Logic for LU Type 6.2*, SC30-3269
- *IBM Systems Network Architecture, Logical Unit 6.2 Reference: Peer Protocols*, SC31-6808
- *Distributed Data Management (DDM) General Information*, GC21-9527

CCSID Publications

- *Character Data Representation Architecture, Executive Overview*, GC09-2207
- *Character Data Representation Architecture Reference and Registry*, SC09-2190

DB2 Server RXXSQL Publications

- *DB2 REXX SQL for VM/ESA Installation and Reference*, SC09-2891

C/370 Publications

- *IBM C/370 Installation and Customization Guide*, GC09-1387
- *IBM C/370 Programming Guide*, SC09-1384

Communication Server for OS/2 Publications

- *Up and Running!*, GC31-8189
- *Network Administration and Subsystem Management Guide*, SC31-8181
- *Command Reference*, SC31-8183
- *Message Reference*, SC31-8185
- *Problem Determination Guide*, SC31-8186

Distributed Database Connection Services (DDCS) Publications

- *DDCS User's Guide for Common Servers*, S20H-4793
- *DDCS for OS/2 Installation and Configuration Guide*, S20H-4795

VTAM Publications

- *VTAM Messages and Codes*, SC31-6493
- *VTAM Network Implementation Guide*, SC31-6494
- *VTAM Operation*, SC31-6495
- *VTAM Programming*, SC31-6496
- *VTAM Programming for LU 6.2*, SC31-6497
- *VTAM Resource Definition Reference*, SC31-6498
- *VTAM Resource Definition Samples*, SC31-6499

CSP/AD and CSP/AE Publications

- *Developing Applications*, SH20-6435
- *CSP/AD and CSP/AE Installation Planning Guide*, GH20-6764
- *Administering CSP/AD and CSP/AE on VM*, SH20-6766
- *Administering CSP/AD and CSP/AE on VSE*, SH20-6767
- *CSP/AD and CSP/AE Planning*, SH20-6770
- *Cross System Product General Information*, GH23-0500

Query Management Facility (QMF) Publications

- *Introducing QMF*, GC27-0714
- *Installing and Managing QMF for VSE*, GC27-0721
- *QMF Reference*, SC27-0715
- *Installing and Managing QMF for VM*, GC27-0720
- *Developing QMF Applications*, SC27-0718
- *QMF Messages and Codes*, GC27-0717
- *Using QMF*, SC27-0716

Query Management Facility (QMF) for Windows Publications

- *Getting Started with QMF for Windows*, SC27-0723
- *Installing and Managing QMF for Windows*, GC27-0722

DL/I DOS/VS Publications

- *DL/I DOS/VS Application Programming*, SH24-5009

COBOL Publications

- *VS COBOL II Migration Guide for VSE*, GC26-3150
- *VS COBOL II Migration Guide for MVS and CMS*, GC26-3151
- *VS COBOL II General Information*, GC26-4042
- *VS COBOL II Language Reference*, GC26-4047

- *VS COBOL II Application Programming Guide*, SC26-4045
- *VS COBOL II Application Programming Debugging*, SC26-4049
- *VS COBOL II Installation and Customization for CMS*, SC26-4213
- *VS COBOL II Installation and Customization for VSE*, SC26-4696
- *VS COBOL II Application Programming Guide for VSE*, SC26-4697

Data Facility Storage Management Subsystem/VM (DFSMS/VM) Publications

- *DFSMS/VM RMS User's Guide and Reference*, SC35-0141

Systems Network Architecture (SNA) Publications

- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *SNA Format and Protocol Reference: Architecture Logic for LU Type 6.2*, SC30-3269
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *SNA Synch Point Services Architecture Reference*, SC31-8134

Miscellaneous Publications

- *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide*, GA32-0100
- *Dictionary of Computing*, ZC20-1699
- *APL2 Programming: Using Structured Query Language*, SH21-1056
- *ESA/390 Principles of Operation*, SA22-7201

Related Feature Publications

- *DB2 for VM Control Center Operations Guide*, GC09-2993
- *DB2 for VSE Control Center Operations Guide*, GC09-2992
- *DB2 Replication Guide and Reference*, SC26-9920

Index

Special characters

- / initialization option 9
- %
 - archive percent 180
 - free space, dbspace 86
 - log cushion percent 187

A

- abnormal end
 - recovery from 147
- access
 - filtered database 182
- access generator tracing 164, 192
- ACCOUNT
 - initialization option 45
 - initialization parameter 193, 195
- accounting
 - ALTACCT command 45
 - ARIACC1 file 45
 - ARIACC2 file 46
 - switching to alternate files 45
 - tape 45
- adapter, online resource 18, 19
- adding
 - dbextent storage 188
 - dbspaces storage 188
- address
 - alternative tape drive 11
 - SHOW ADDRESS command 69
 - SHOW ADDRESS example 70
 - showing address for modules 69
- adjusting bias of dispatcher 181
- AG trace 164, 192
- agent 65
 - checkpoint 66
 - definition 15
 - description 65
 - determining agent number 51, 66
 - displaying status 66
 - forcing agents 67
 - generating 184
 - inactive users 136
 - operator agent 65
 - ready/recovery agent 65
 - tracing 192, 193
- agent identifier 66
- agent number 66
- agent structure
 - checkpoint 66
 - definition 15
 - description 65
 - determining agent number 51, 66
 - displaying status 66
 - forcing agents 67
 - generating 184
 - inactive users 136
 - operator agent 65
 - ready/recovery agent 65

- AGENT trace 192, 193
- ALT option of FILEDEF command 11
- ALTACCT command 46
- alternate accounting files 45
- alternative tape drive, virtual device
 - number 11
- APPCVM 17
- APPL (batch application)
 - definition 66
- application program, FORCE
 - command 53
- application requester 8
- application server 2
- archive
 - ARCHIVE operator command 46, 47
 - automatic 183
 - database archive 12
 - LARCHIVE command 50
 - log archive 12
 - log archive to disk 51
 - log archive to tape 50
 - log cushion percent 187
 - log size 113
 - online database archive 47
 - output file (ARIARCH) 10
 - percent 180
 - specifying restore at startup 188
 - SQLEND UARCHIVE command 48
 - summary 14
 - taking an archive at shutdown 12
 - user archive 12, 48
 - verifying the directory 13
 - VSE/VSAM user archive 49
- ARCHPCT
 - initialization option 113
 - initialization parameters 180
 - log size 113
- ARCHTAPE 193
 - initialization parameter 193
- ARI0087D prompt 163
- ARIACC1 accounting file 45
- ARIACC2 accounting file 46
- ARIARCH, database archive file
 - output 10
- ARILARC, log archive file output 10
- ARIS080D procedure 19
- arithmetic operator
 - in syntax diagrams ix
- ARITRAC, tracing file output 10
- AU trace 192
- auditing security
 - tracing 164
- authorization ID
 - CHECKPT 94, 106, 109
- authorization tracing 164, 192
- automatic archive 183

B

- BEGINLUW counter 58
- beginning online support 18

- bias of dispatcher, adjusting 181
- blanks, initialization options 4
- BLKSIZE option
 - FILEDEF commands 10
- block dump, specifying 181
- block size, FILEDEF commands 10
- blocks
 - number of lock requests 185
- buffer
 - data pages 186
 - number of directory buffers 184
 - SHOW BUFFERS command 70
 - SHOW BUFFERS example 71

C

- call tracing 163, 191
- catalog tables
 - generation 188
 - index reorganization 188
 - migration 188
- cataloging a source member 6
- CCSID conversion tracing 193
- change settings, initialization
 - parameters 63
- character set
 - specifying 180
- CHARNAME initialization
 - parameters 180
- checkpoint
 - agent 66
 - counter 58
 - interval 180
 - log archive 68, 69
 - logical unit of work 68, 69
- CHECKPT authorization ID 94, 106, 109
- CHKINTVL 180
- CHKPOINT counter 58
- choose
 - in syntax diagrams ix
- CICS 18
 - CICS/VS monitoring facility 57
 - CICS/VSE v
 - DBDCCICS 19
 - in-doubt logical units of work 152
 - restart resynchronization facility 152
 - terminal 18, 44
 - transactions that access the application server 20
- CICSUSER option, CIRA transaction 34
- CICSUSER option, CIRB transaction 19
- CIRB transaction
 - CICSUSER option 18
 - description 18
- CIRC transaction 36
- CIRD transaction 20
 - example 23, 24, 25, 26, 27, 28
- CIRR transaction 26
- CIRT transaction 29
- CLOSE option of TRACE command 165
- closing a trace file 166

CMS
 FILEDEF commands for DB2 Server
 for VM tracing 166
 files, initialization options 5
 files, initialization parameters 186
 CODE trace 164
 cold start 188
 COM trace 192
 command
 operator 199
 command, operator
 ALTACCT 45
 ARCHIVE 46
 COUNTER 58
 COUNTER POOL 199
 FORCE 52, 67
 FORCE RMTUSER 54
 issuing 43
 LARCHIVE 50
 RESET 61
 RESET CRR LOGNAMES 62
 RESET HIGHSTOR 62
 RESET INDOUBT 63
 RESET POOL 202
 SET 63
 SET APPCVM 17
 SET LANGUAGE 56
 SET POOL 203
 SET SAVEINTV 205
 SET TARGETWS 206
 SHOW ACTIVE 65, 206
 SHOW ADDRESS 69
 SHOW BUFFERS 70, 207
 SHOW CONNECT 73
 SHOW CRR LOGNAMES 80
 SHOW DBCONFIG 81
 SHOW DBEXTENT 84
 SHOW DBSPACE 85
 SHOW INDOUBT 87
 SHOW INITPARM 89
 SHOW INVALID 90
 SHOW LOCK ACTIVE 93, 208
 SHOW LOCK DBSPACE 96
 SHOW LOCK GRAPH 98, 209
 SHOW LOCK MATRIX 103
 SHOW LOCK USER 106
 SHOW LOCK WANTLOCK 108
 SHOW LOG 50, 113
 SHOW LOGHIST 114
 SHOW POOL 118, 209
 SHOW SAVEINTV 211
 SHOW STORAGE 130
 SHOW SYSTEM 133
 SHOW TARGETWS 212
 SQLEND 11
 syntax viii
 TRACE 161
 commas, initialization options 4, 6
 comments, CMS files for starting the
 application server 5
 comments, source members for starting
 the application server 6
 committing changes
 COMMIT option of FORCE
 command 53
 communication
 tracing 192

communication (*continued*)
 wait 102
 concurrent
 scans 184
 users 184
 configuration
 application server 81
 connected users, displaying 122, 125,
 135, 137, 138, 141, 143
 console
 disconnecting 11
 how to access 43
 issuing commands 44
 issuing operator commands 43
 SHOW PROC command 122
 SHOW PSERVER command 125
 SHOW USERS command 135
 signing off 11
 START PROC command 137
 START PSERVER command 138
 START TCPIP command 140
 STOP PROC command 141
 STOP PSERVER command 143
 control
 terminal
 CICS 44
 control block dump, specifying 181
 Control Program (CP)
 CP DISCONN command 11
 control statements
 for trace formatter 170
 CONTROLPW password 194
 CONV trace 193
 conventions
 syntax diagram notation viii
 Conversational Monitor System (CMS)
 work units 136
 conversion, lock 101, 111
 Coordinated Resource Recovery 150
 counter
 storage pool performance
 Data Spaces Support 200
 standard DASD I/O system 199
 COUNTER operator command 58
 COUNTER POOL 199
 counters
 BEGINLUW 58
 CHKPOINT 58
 DASDIO 59
 DASDREAD 59
 DASDWRT 59
 DBSSCALL 58
 DEADLCK 59
 DIRREAD 59
 DIRWRITE 59
 ESCALATE 59
 LDIRBUFF 59
 LOCKLMT 58
 LOGREAD 59
 LOGWRITE 59
 LPAGBUFF 59
 LTIMEOUT 59
 PAGEREAD 59
 PAGWRITE 59
 RDSCALL 58
 resetting 61, 62
 ROLLBACK 58

counters (*continued*)
 WAITLOCK 59
 CP DUMP 164
 CRR
 CRR commands 150
 displaying in-doubt agents at the
 database manager 87
 displaying log names DB2 uses during
 resynchronization initialization 80
 erasing records of in-doubt LUWs
 forced by the FORCE command 63
 FORCE RINIT for forcing
 Resynchronization Initialization 54
 FORCE RREC for forcing
 Resynchronization Recovery 54
 resetting log names DB2 uses during
 Resynchronization Initialization 62
 resynchronization initialization status
 displayed using SHOW
 CONNECT 79
 resynchronization recovery status
 displayed using SHOW
 CONNECT 79
 cushion
 log 113, 187

D

DASDIO counter 59
 DASDREAD counter 59
 DASDWRT counter 59
 data control tracing 163, 191
 data conversion (CONV)
 tracing 193
 data manipulation
 tracing 191
 data manipulation, tracing 163
 data page buffers
 specifying number 186
 data stream trace 164, 172
 Data System Control (DSC) 160, 163
 list of functions and
 subcomponents 164
 database 84
 application server 8
 database archive output file
 (ARIARCH) 10
 database extents 82, 84
 DB2 Server for VSE & VM database
 manager operator v
 filtered access 182
 generation 188
 machine operator 44
 mode 195
 online archive 46, 47
 showing application server
 configuration 81
 showing database extents 84
 showing database spaces 85
 showing dbspace locks 96
 specifying name 194
 specifying password 194
 storage pool 118
 database archive
 online 46
 database archive, online 47
 database configuration 127

Database Storage Subsystem (DBSS) 160, 163
 calls 58
 functions and subcomponents 163
 DAXP transaction 39
 DAXT transaction 41
 DB2 Server for VSE & VM
 definition 1
 disconnecting 11
 FILEDEF commands 10
 forcing a user 51
 initialization options 3
 initialization parameters 179
 normal shutdown 14
 online support parameters 17
 online support parameters for VSE
 guest sharing 17
 operating 43
 operator v, 1, 43
 partition 1, 8
 starting
 error messages 4
 multiple user mode 1, 8
 single user mode 8
 stopping 11
 trace facility 155
 trace-point output 174
 tracing a CMS file 158
 tracing in multiple user mode 157
 tracing in single user mode 158
 VSE guest sharing 17
 VSE Guest Sharing v
 DB2 Server for VSE & VM operator console
 disconnecting 11
 how to access 43
 issuing commands 44
 SHOW PROC command 122
 SHOW PSERVER command 125
 SHOW USERS command 135
 START PROC command 137
 START PSERVER command 138
 START TCPIP command 140
 STOP PROC command 141
 STOP PSERVER command 143
 DB2 Server for VSE and VM CICS support, online support 17
 DB2 Server for VSE CICS support, online support 17
 DB2 Server for VSE operator console, signing off 11
 dbextent storage
 adding to a database 188
 deleting from a database 188
 description 84
 DBMODE initialization parameter 195
 DBNAME initialization parameter 194
 DBPSWD initialization parameter 194
 dbspace storage
 adding 188
 description 83
 displaying page usage 85
 percentage of free space 86
 show dbspace name 96
 show dbspace number 85
 show locks 96
 DBSPACENAME 96
 DBSPACENO 85
 DBSPNUM initialization parameter 180
 DBSS
 calls 58
 functions and subcomponents 163
 DBSS (Database Storage Subsystem) 160, 163
 DBSSCALL counter 58
 DC trace 163, 191
 DCSSID option 3
 DDR (Dump Restore Service Program) 48
 DEADLCK counter 59
 deadlock, description 59
 default
 in syntax diagrams x
 initialization options 2, 3
 default server_name 26
 defuid option
 with the CIRA transaction 34
 with the CIRB transaction 19
 delaying a checkpoint, log archive 69
 delaying a log archive checkpoint 68
 deleting
 dbextent storage 188
 determining problems 155
 DICT trace 191
 dictionary tracing 191
 directory
 number of directory buffers 184
 verifying 13
 DIRREAD counter 59
 DIRWRITE counter 59
 DISABLE option, FORCE command 53
 DISCONN command 11
 disconnecting from the DB2 machine 11
 discontinuous saved segments 3
 DISK
 option of TRACE command 161
 disk failure recovery 154
 disk, log archive 51
 dispatcher, adjusting bias 181
 DISPBIAS initialization parameter 181
 display
 checkpoint performance data 181
 display current settings 89
 display device, specifying 194
 display, issuing operator commands 45
 displaying status information 20
 Distributed Relational Resource Manager (DRRM) 160
 DLBL statement 2, 8
 DM trace 163, 191
 DRDA TRUE support
 using CICS DAXP transaction 39
 using CICS DAXT transaction 41
 DRDA2 command
 RESET CRR LOGNAMES 62
 resetting log names DB2 uses during Resynchronization Initialization 62
 DRDA2 commands
 DAXP 39
 DAXT 41
 disabling DRDA2 support in CICS 41
 displaying in-doubt agents at the database manager 87
 DRDA2 commands (*continued*)
 displaying log names DB2 uses during resynchronization initialization 80
 displaying status of DRDA2 distributed unit of work support (SYNCPNT parameter) 89
 erasing records of in-doubt LUWs forced by the FORCE command 63
 FORCE RINIT for forcing Resynchronization Initialization 54
 FORCE RREC for forcing Resynchronization Recovery 54
 managing DRDA2 support in CICS 38
 RESET INDOUBT 63
 setting parameter defaults used when DRDA2 support is enabled in CICS 39
 SHOW CONNECT and resynchronization initialization 79
 SHOW CONNECT and resynchronization recovery 79
 SHOW CONNECT protocol indication 73
 SHOW CRR LOGNAMES 80
 SHOW INDOUBT 87
 SYNCPNT startup parm 189
 DRDA2 Commands
 CRR commands 150
 using SHOW CONNECT and SHOW INDOUBT 148
 drive, alternative tape 11
 DRRM (Distributed Relational Resource Manager) 160
 DSC (Data System Control) 160, 163
 list of functions and subcomponents 164
 DSPLYDEV
 initialization options 7
 initialization parameter 194
 DSPSTATS initialization parameter 181
 dual log 181
 DUALLOG initialization parameter 181
 dump
 CP
 TRACE CLOSE option 165
 TRACE NOCLOSE option 165
 specifying dump types 181
 DUMP
 CP
 snapshot dump 164
 option of TRACE command 161
 Dump Restore Service Program (DDR) 48
 dump, minidump 147
 DUMPALL trace formatter option 168
 dumps
 snapshot 162
 DUMPTYPE initialization parameter 181
 DVERIFY 13
 dynamic assign, archive 47

E

empty pages
 releasing 180, 188

- ending
 - online support 29
 - the DB2 session 11
- entry call tracing 163, 191
- ENTRY trace 163, 191
- environments, operating v
- error messages 4
- ESCALATE counter 59
- escalation
 - description 58
 - lock 185
- EXEC
 - invoking 166
 - SQLSTART 2
 - SQLTRFMT 166
- EXEC trace 192, 193
 - RDS 164
 - executives 164
 - functions and subcomponents 164
- executing online support 18
- EXIT trace 163, 191
- EXTEND initialization parameter 182
- extents, database
 - SHOW DBCONFIG command 82
 - SHOW DBEXTENT command 84
- extents, database, SHOW DBEXTENT command 84
- extents, database, SHOW SQLDBGEN command 127

F

- facility
 - CICS restart resynchronization 152
 - CICS/VS monitoring 57
 - trace 155
- failure, recovery 147
 - disk failures 154
 - general 147
- fast application server
 - online support shutdown 30
 - shutdown 15
- file parameter, example 5
- FILEDEF command
 - ALT option 11
 - ARIARCH 10
 - ARILARC 10
 - ARITRAC 10
 - block size option 10
- files, CMS initialization options 5
- filtered
 - database access 182
 - log recovery 182
- First Failure Data Capture support 147
- FORCE operator command
 - COMMIT option 53
 - DISABLE option 53
 - effect on application programs 53
 - forcing users to end work 51
 - ROLLBACK option 53
 - syntax 52
 - VSE guest sharing 53, 67
- FORCE RMTUSER operator command
 - forcing remote users to end work 53
 - syntax 54
- forceable agents 67

- forcing
 - DB2 Server for VSE remote user 53
 - in-doubt logical units of work 153
 - user 51
- formatting
 - database 188
 - trace output 166, 168
- fragment of syntax
 - in syntax diagrams xi
- free space, dbspace 86
- full &part. dump, specifying 181

G

- GEN trace 191
- general recovery 147
- generating agent structures 184
- generator tracing
 - access generator 164
 - DBSS subsystem 191
 - statement generator 164, 192
- graph
 - SHOW LOCK GRAPH command 98
 - SHOW LOCK GRAPH example 102, 103
- guest sharing, VSE v, 17

H

- halting the application server
 - description 11
 - online support 29
- hardware failure recovery 154
- heuristic decisions
 - in-doubt LUW 87
 - in-doubt LUWs 63
- host variable
 - in syntax diagrams ix

I

- identifier
 - logical unit of work 52
 - parameter 186
- immediate application server
 - online support shutdown 30
 - shutdown 15
- in-doubt logical units of work
 - definition 152
 - description 67
- in-doubt LUW 87
- in-doubt LUWs 63
- index
 - DBSS tracing 163
 - reorganization 188
 - show invalid indexes 90
- INDEX trace 163, 191
- initialization parameters
 - ACCOUNT 193, 195
 - ARCHPCT 180
 - change settings 63
 - CHARNAME 180
 - CHKINTVL 180
 - CMS files 5, 186
 - DBMODE 195
 - DBNAME 194

initialization parameters (*continued*)

- DBPSWD 194
- DBSPNUM 180
- DISPBIAS 181
- display current settings 89
- DSPLYDEV 7, 194
- DSPSTATS 181
- DUALLOG 181
- DUMPTYPE 181
- EXTEND 182
- list 179
- LOGMODE 183
- MAPPING 196
- NCSCANS 184
- NCUSERS 184
- NDIRBUF 184
- NLRBS 185
- NLRBU 185
- NPACKAGE 185
- NPACKPCT 185
- NPAGBUF 186
- overriding 6
- PARMID 186
- PROCXMAB 186
- PROGNAME 186
- PROTOCOL 196
- PTIMEOUT 186
- RMTUSERS 194
- SAVEINTV 196
- SECALVER 189
- SECTYPE 189
- SEPINTDB 197
- SLOGCUSH 187
- SOSLEVEL 187
- source member 5
- specifying 3
- STARTUP 188
- SYNCPNT 189
- SYSMODE 189
- TARGETWS 197
- TCPCAN 196
- TCPDISPB 195
- TCPPORT 190
- TRACCONV 193
- TRACDBSS 160, 191
- TRACDRRM 191
- TRACDSC 160, 192
- TRACEBUF 192
- TRACRDS 160, 192
- TRACSTG 192
- TRACWUM 193
- initializing
 - online support 18
- input parameters 179
- INT trace 164, 192
- intention exclusive (IX) mode 85
- interpreter tracing 164, 192
- interval checkpoint 180
- interval option
 - with the CIRR transaction 35
 - with the CIRT transaction 31
- invalid index, SHOW INVALID example 93
- isolating problems 155
 - with ISQLMAP 177
 - with ISQLTRACE 177
- ISQL user data, CIRD transaction 22

ISQLTRACE command 177
IX (intention exclusive) mode 95

J

JCL (job control)
format trace output 168
source members 6
starting the application server 1
starting the application server in
single user mode 8
user archive 49
job control (JCL)
format trace output 168
source members 6
starting the application server 1
starting the application server in
single user mode 8
user archive 49

K

keyword
in syntax diagrams viii

L

langid option
CIRB transaction 20
SET LANGUAGE command 56
language
choosing character sets 180
CIRB transaction, langid option 20
SET LANGUAGE operator
command 56
LARCHIVE operator command 50
LDIRBUFF counter 59
leaving the application server 11
level 0 trace 163
level 1 trace 163
level 2 trace 163
lock 58
conversion 101, 111
counter 58
escalation 58, 185
modes 102
request mode 111
request state 111
show active locks 93
SHOW LOCK DBSPACE 96
SHOW LOCK GRAPH command 98
SHOW LOCK GRAPH example 102,
103
SHOW LOCK MATRIX
command 103
SHOW LOCK USER command 106
SHOW LOCK WANTLOCK
command 108
lock management tracing 163, 191
lock request block (LRB)
NLRBS initialization parameter 185
number per user 185
usage 103
LOCK trace 163, 191
lock wait table 104
LOCKLMT counter 58

log
archive
at shutdown 12
definition 50
to disk 51
to tape 50
archive file output (ARILARC) 10
cushion 113, 187
data sets 188
determining size 113
dual logging 181
filtered log recovery 182
log usage 113
LOGWRITE counter 59
management tracing 163, 191
online log archive, LARCHIVE
command 50
SHOW LOG 113
specifying
log mode 183
trace 191
two logs 181
trace 163
logging off the application server 11
logical unit of work (LUW)
counter 58
delaying a checkpoint 68, 69
example 67
identifier 52, 66
in-doubt 67, 152
lock management 163
multiple 136
LOGMODE initialization parameter 183
LOGREAD counter 59
LOGWRITE counter 59
LPAGBUFF 59
LRB (lock request block)
NLRBS initialization parameter 185
number per user 185
usage 103
LTIMEOUT 59
LTIMEOUT initialization parameter 184

M

machine, virtual
database 8
user 8
manually resolving in-doubt logical units
of work 153
MAPPING
initialization parameter 196
MASTERPW password 194
matrix, SHOW LOCK MATRIX
command 103
media recovery 154
members, source 5
A-type 5
cataloging example 6
messages
error 4
national language 20, 56
minidump 147
mode option, CIRB transaction 35
modes
CIRT transaction 30
database 195

modes (continued)
lock 102
multiple user mode 8
single user mode 8
module address 69
monitoring
CICS/VS monitoring facility 57
dbspace page consumption 85
multiple logical units of work 136
multiple logs, specifying 181
multiple user mode 8

N

name
server-name 194
national language
character set
choosing 180
online support 20
SET LANGUAGE operator
command 56
NCSCANS initialization parameter 184
NCUSERS initialization parameter 184
NDIRBUF initialization parameter 184
NLRBS initialization parameter 185
NLRBU initialization parameter 185
NOCLOSE option of TRACE
command 165
nolinks option
CIRA transaction 33
CIRB transaction 19
nonrecoverable storage pool
description 84, 121
normal application server
online support shutdown 30
shutdown 14
NPACKAGE initialization
parameter 185
NPACKPCT initialization parameter 185
NPAGBUF initialization parameter 186
null id string 66

O

online
log archive 50
status information 20
online resource adapter 31, 32, 36
online support
CIRB transaction options 18
definition 1
monitoring 20
online resource adapter 18
resource adapter 19
starting 18
stopping 29
VSE guests 17
online support parameters 18, 38
operating environments v
operating systems v
operating the application server 1, 43
operator
application server 43
DB2 Server for VSE & VM system v
operator agent 65

- operator command
 - ALTACCT 45
 - ARCHIVE 46
 - COUNTER 58
 - FORCE 52, 67
 - FORCE RMTUSER 54
 - issuing 43
 - LARCHIVE 50
 - RESET 61
 - RESET CRR LOGNAMES 62
 - RESET HIGHSTOR 62
 - RESET INDOUBT 63
 - SET 63
 - SET APPCVM 17
 - SET LANGUAGE 56
 - SHOW ACTIVE 65
 - SHOW ADDRESS 69
 - SHOW BUFFERS 70
 - SHOW CONNECT 73
 - SHOW CRR LOGNAMES 80
 - SHOW DBCONFIG 81
 - SHOW DBEXTENT 84
 - SHOW DBSPACE 85
 - SHOW INDOUBT 87
 - SHOW INITPARM 89
 - SHOW INVALID 90
 - SHOW LOCK ACTIVE 93
 - SHOW LOCK DBSPACE 96
 - SHOW LOCK GRAPH 98
 - SHOW LOCK MATRIX 103
 - SHOW LOCK USER 106
 - SHOW LOCK WANTLOCK 108
 - SHOW LOG 50, 113
 - SHOW LOGHIST 114
 - SHOW POOL 118
 - SHOW STORAGE 130
 - SHOW SYSTEM 133
 - SQLEND 11
 - syntax viii
 - TRACE 161
- operator commands 199
- operator console system, issuing operator commands 43
- OPT trace 164, 192
- optimizer
 - tracing 164, 192
- optional
 - default parameter
 - in syntax diagrams xi
 - item
 - in syntax diagrams ix
 - keyword
 - in syntax diagrams xi
- ordinary application server
 - online support shutdown 30
 - shutdown 14

P

- PA trace 164, 192
- page buffers, data 186
- page consumption, dbspace 85
- page usage, dbspace 85
- PAGEREAD counter 59
- PAGWRITE counter 59
- parameter
 - DCSSID 3

- parameter (*continued*)
 - identifier 186
 - initialization 3, 179
 - overriding 6
 - initialization in CMS files 186
 - initialization, CMS files 5
 - PARM 4
 - user program 9
- parameter file, example 5
- parentheses
 - in syntax diagrams ix
- PARM option 4
- PARMID initialization parameter 186
- PARSE trace 191
- parser tracing 164, 191, 192
- partial dump 181
- partition
 - application server 1
 - DB2 8
 - dump, specifying 181
 - single partition mode
 - See single user mode
 - user 8
- password
 - specifying database 194
 - VSAM 194
- password implications 32
- password option
 - CIRB transaction 18
 - CIRR transaction 34
 - CIRT transaction 30
- PDUMP
- TRACE
 - CLOSE option 164
 - NOCLOSE option 164
- percent
 - archive percent 180
 - free space, dbspace 86
 - log cushion percent 187
- performance monitoring 57
- pools storage
 - description 118
 - nonrecoverable 84, 121
 - SHOW POOL example 118, 119
 - showing 118
- problem
 - determination 155
- procedures, ARIS080D 19
- PROCMXAB initialization parameter 186
- PROGNAME initialization parameter 186
- program
 - name 186
 - user
 - options 9
 - single user mode 9
- PROTOCOL initialization parameter
 - description 196
- PROTOCOL parameter
 - for initialization 196
- PTIMEOUT initialization parameter 186
- punctuation mark
 - in syntax diagrams ix
- purging the trace buffer 13

Q

- quick application server
 - online support shutdown 30
 - return code 15
 - shutdown 15
- quick DB2 system
 - shutdown 13
 - SQLEND QUICK 13

R

- RDIIN trace 191
- RDS (Relational Data System) 163
 - calls 58
 - list of functions and subcomponents 164
 - tracing 163
- RDSCALL counter 58
- read counters 59
- ready/recovery agent 65
- recovery
 - general 147
 - management tracing 191
 - media failure 154
 - recovery management tracing 163
- redefining the log data sets 188
- reformatting, database 188
- regular application server
 - online support shutdown 30
 - shutdown 14
- Relational Data System (RDS) 163
 - calls 58
 - list of functions and subcomponents 164
 - tracing 163
- reorganizing the catalog index 188
- repeat symbol
 - in syntax diagrams x
- request
 - blocks 185
 - mode for locks 111
 - state for locks 111
- required item
 - in syntax diagrams ix
- reserved words
 - SQL xii
- RESET CRR LOGNAMES operator
 - command 62
- RESET HIGHSTOR operator
 - command 62
- RESET INDOUBT command 63
- RESET operator command 61
- RESET POOL command
 - description 202
- resetting
 - counters 61, 62
- resetting log names 62
- resolving in-doubt logical units of work 153
- resource adapter
 - online, description 18
 - rmid (resource adapter ID), CIRB transaction 19
- restart resynchronization
 - facility 152

- restoring data
 - archive tape 188
- resynchronization
 - facility, CICS 152
- returns from DBSS entry calls 163, 191
- rewinding tapes 161
- rmid option, CIRB transaction 19
- RMTUSERS parameter
 - for initialization 194
- ROLLBACK
 - counter 58
 - option of FORCE command 53
- routing
 - the display of parameters 194

S

- save interval
 - SET 205
 - SHOW 211
- saved segment 3
- SAVEINTV initialization parameter 196
- scans, concurrent 184
- SCIF facility 44
- SECALVER initialization parameter
 - description 189
- SECALVER parameter
 - for initialization 189
- SECTYPE initialization parameter
 - description 189
- SECTYPE parameter
 - for initialization 189
- security
 - audit tracing 164, 192
 - auditing 164
- SELECT DBSPACENO statement 85
- SEPINTDB initialization parameter 197
- server_name option
 - CIRA transaction 34, 35
 - CIRB transaction 20
 - CIRC transaction 36
 - CIRD transaction 21
- SET APPCVM 17
- SET command 63
- SET LANGUAGE operator command 56
- SET POOL 203
- SET SAVEINTV 205
- SET TARGETWS 206
- SG trace 164, 192
- sharing, guest v, 17
- short on storage
 - flag 85, 121
 - level 187
- SHOW ACTIVE 206
- SHOW ACTIVE command
 - description 65
 - display
 - NEW 66
 - NIW 66
 - R/O 66
 - R/W 66
 - example 68
 - forcing 51
 - forcing remote users to end work 53
- SHOW ADDRESS command 69
- SHOW BUFFERS 207
- SHOW BUFFERS command 70

- SHOW command 135
- SHOW commands 65, 133
- SHOW CONNECT
 - command 73
 - forcing remote users to end work 53
 - forcing users to end work 51
 - for CRR in VM/ESA 79
 - for Stored Procedures 79
- SHOW CRR LOGNAMES command 80
- SHOW DBCONFIG command 81
- SHOW DBEXTENT
 - command 84
- SHOW DBSPACE
 - command 85
- SHOW INDOUBT command 87
- SHOW INITPARM command 89
- SHOW INVALID
 - command 90
- SHOW LOCK
 - ACTIVE command 93
 - DBSPACE command 96
 - GRAPH command 98
 - MATRIX command 103
 - USER command 106
 - WANTLOCK command 108
- SHOW LOCK ACTIVE 208
- SHOW LOCK GRAPH 209
- SHOW LOG command 113
- SHOW LOGHIST command 114
- SHOW POOL 209
- SHOW POOL command 118
- SHOW PROC command 122
- SHOW PSERVER command 125
- SHOW SAVEINTV 211
- SHOW SQLDBGEN
 - command 127
- SHOW STORAGE command 130
- SHOW SYSTEM command 133
- SHOW TARGETWS 212
- SHOW USERS command 135
- shutting down the application server
 - archives 14
 - description 11
 - normal shutdown 14
 - online support 29
- signing off the DB2 Server for VSE
 - system console 11
- Single Console Image Facility (SCIF) 44
- single partition mode
 - See single user mode
- single user mode
 - description 8
 - initialization parameter 189
 - program name 186
 - the application server 8
 - user program 9
- slash (/) initialization option 9
- SLOGCUSH initialization option
 - log size 113
- SLOGCUSH initialization parameter
 - description 187
- snapshot dump 162
- SORT trace 163, 191
- SOSLEVEL
 - initialization option 121
 - initialization parameter 187

- source members 5
 - A-type 5
 - cataloging example 6
- SQLDBAPW password 30
- SQLEND operator command
 - ARCHIVE 12
 - description 11
 - DVERIFY 13
 - LARCHIVE 12
 - QUICK 13
 - syntax 11
 - TRCPURGE 13
 - UARCHIVE 12, 48
- SQLSTART EXEC
 - parameter options 195
 - syntax diagram 2
- SQLTRFMT EXEC control
 - statements 166
- START PROC command 137
- START PSERVER command 138
- START TCPIP command 140
- starting the application server
 - single user mode 8
- starting the database manager
 - after a user archive 188
 - error messages 4
 - initialization parameters 179
 - multiple user mode 1, 8
 - online support 18
 - single user mode 8, 9
- starting the DB2 Server for VM machine
 - multiple user mode 8
 - single user mode 8
- STARTUP initialization parameter 188
- STAT trace 163, 191
- statement generator
 - tracing 164, 192
- static assign, archive 47
- statistics
 - DBSS update statistics tracing 163
 - tracing 191
- STG (Storage) 160
- STOP PROC command 141
- STOP PSERVER command 143
- stopping
 - application server
 - archives 14
 - description 11, 29
 - online support 29
- STOR trace 163, 191
- storage
 - resetting 62
 - short on storage level 187
 - virtual 1
- Storage (STG) 160, 163
- storage management
 - tracing 163, 191
- storage pool
 - description 118
 - nonrecoverable 84, 121
 - SHOW POOL example 118, 119
 - showing 118
- storage, short on storage level 187
- storage, SHOW STORAGE
 - command 130
- SUBS (subsystem)
 - definition 66

- subsystem
 - calls to DBSS 58
 - DBSS update statistics tracing 163
 - tracing CONV (CCSID and Data Conversion) 163
 - tracing DBSS (Database Storage Subsystem) 163
- summary
 - of archives 14
- support, online
 - CIRB transaction options 18
 - definition 1
 - resource adapter 18, 19
 - starting 18
 - VSE guests 17
- switching accounting files 45
- sync point manager (SPM) 189
 - disabling 189
- SYNCPNT initialization parameter 189
- syntax diagram
 - notation conventions viii
- syntax diagram, SQLSTART EXEC 2
- SYSMODE initialization parameter 189
- system
 - console, signing off 11
 - SHOW SYSTEM command 133
- system mode, specifying 189
- system operator console, issuing operator commands 43

T

- tape
 - accounting 45
 - log archive 50
 - rewinding 161
- tape drive, alternative virtual device number 11
- TAPEMGR 190
 - initialization parameter 190
- target working storage size
 - SET 206
 - SHOW 212
- TARGETWS initialization parameter 197
- task-related user exits
 - disabling DRDA support 41
 - enabling 39
 - managing 38
 - setting parameters 39
- Task-Related User exits 38
- TASKNO (CICS task number) 22
- TCPCAN parameter
 - for initialization 196
- TCPDISPB parameter
 - for initialization 195
- TCPMAXRT 190
 - initialization parameter 190
- TCPPOINT initialization parameter
 - description 190
- TCPPOINT parameter
 - for initialization 190
- TCPRETRY 191
 - initialization parameter 191
- TERMID (CICS terminal ID) 22
- terminal control, CICS 44
- terminal, CICS 18

- terminating the application server
 - online support 29
 - SQLEND operator command 11
- TLBL statement 8
- TRACCONV initialization
 - parameter 193
- TRACDBSS initialization parameter 160, 191
- TRACDRRM initialization
 - parameter 191
- TRACDSC initialization parameter 160, 192
- trace buffer
 - purging 13
- trace DBSS, specifying 191
- TRACE operator command
 - DISK option 161
 - DUMP option 161
- trace services 166, 168
- TRACEBUF initialization parameter 192
- tracing
 - choosing subcomponents 159
 - closing a trace file 166
 - DBSS (Database Storage Subsystem) 163, 191
 - DBSS functions and subcomponents 163
 - DRDA data stream 164, 172
 - DRRM (Distributed Relational Resource Adapter) 163
 - DSC (Data System Control) 163
 - DSC functions and subcomponents 164
 - facility 155
 - formatting utility 166, 168
 - level 0 trace 163
 - level 1 trace 163
 - level 2 trace 163
 - output file (ARITRAC) 10
 - RDS (Relational Data System) 163
 - starting a trace 160
 - STG (Storage) 163
 - stopping a trace 165
 - to tape 167
 - WUM (Work Unit Manager) 163
- TRACRDS initialization parameter 160, 192
- TRACSTG initialization parameter 192
- TRACWUM initialization parameter 193
- TRANID (CICS transaction ID) 22
- transactions
 - CIRA 33
 - CIRB 18
 - CIRC 36
 - CIRD 20
 - CIRR 34
 - CIRT 29
 - DAXP 39
 - DAXT 41
- transient index 90, 93
- TRCPURGE 13
- TRUE (Task-Related User exits) 38
- TRUE (task-related user exits)
 - disabling 41
 - TRUE (task-related user exits)
 - enabling 39

- turning off the application server
 - online support 29
 - SQLEND operator command 11
- turning on online support 18
- two logs, specifying 181

U

- update statistics tracing 163, 191
- user
 - forcing 51
 - partitions 8
 - SHOW LOCK USER command 106
 - SHOW PROC command 122
 - SHOW PROC example 124
 - SHOW PSERVER command 125
 - SHOW PSERVER example 126
 - SHOW USERS command 135
 - SHOW USERS example 136
 - START PROC command 137
 - START PROC example 138
 - START PSERVER command 138
 - START PSERVER example 140
 - START TCPIP command 140
 - STOP PROC command 141
 - STOP PROC example 142
 - STOP PSERVER command 143
 - STOP PSERVER example 145
- user archive
 - creating 48
 - database manager start-up 188
 - UARCHIVE option 12
 - VSE/VSAM example 49
- user display, issuing operator commands 45
- user ID
 - CICS 18
 - virtual machine 66, 93, 136
- user mode
 - multiple 8
 - single 8
- user program
 - options 9
 - single user mode 9
- user, remote
 - forcing 53
- USERDATA 22
- USERID, CIRD transaction 22
- users, number of concurrent 184

V

- verifying
 - the directory 13
- virtual device number, alternative tape drive 11
- virtual machine
 - database 8
 - DB2 Server for VM 8
 - user 8
- virtual storage 1
- VM system console, issuing operator commands 43
- VM/ESA v

- VM/ESA (Virtual Machine/Enterprise Systems Architecture)
 - environment v
- VSAM
 - BACKUP command 49
 - password
 - CONTROLPW 194
 - MASTERPW 194
- VSE (Virtual Storage Extended)
 - guest sharing v, 17
- VSE/Advanced Functions v
- VSE/VSAM BACKUP command
 - example 49

W

- wait, communication 102
- WAITLOCK counter 59
- WANTLOCK, SHOW LOCK
 - WANTLOCK command 108
- warm start 188
- without DISABLE option, FORCE
 - command 53
- Work Unit Manager (WUM) 160
- work units
 - CMS 136
 - multiple 136
- write counters 59
- WUM (Work Unit Manager) 160

Contacting IBM

Before you contact DB2 customer support, check the product manuals for help with your specific technical problem.

For information or to order any of the DB2 Server for VSE & VM products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product information

DB2 Server for VSE & VM product information is available by telephone or by the World Wide Web at <http://www.ibm.com/software/data/db2/vse-vm>

This site contains the latest information on the technical library, product manuals, newsgroups, APARs, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at <http://www.ibm.com/planetwide>

In some countries, IBM-authorized dealers should contact their dealer support structure for information.



File Number: S370/4300-50
Program Number: 5697-F42

Printed in USA

SC09-2986-03



Spine information:



DB2 Server for VSE & VM

Operation

Version 7 Release 5