DB2 Universal Database for OS/390

Reference for Remote DRDA Requesters and Servers

Version 6

SC26-9012-00

Note!

Before using this information and the product it supports, be sure to read the general information under Appendix A, “Notices” on page 45.

First Edition (June 1999)

This edition applies to Version 6 of DB2 Universal Database Server for OS/390, 5645-DB2, and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

Contents

Introduction

This book explains:

- What DDM commands DB2® for OS/390® uses. For each DDM command, a table summarizes whether the DB2 application requester (AR) and the DB2 application server (AS) support the command parameters.
- How an accounting system can track DRDA® access to DB2.
- How an AR can request data from a DB2 data sharing group.

Who should read this book

This book is intended for people who are implementing an application requester (AR) or an application server (AS) that communicates with DB2 at the first, second, or third levels of Distributed Relational Database Architecture™ (DRDA). This can be either:

- A DRDA AR that connects to a DB2 AS, or
- A DRDA AS that services applications at a DB2 DRDA AR.

The fourth level of DRDA adds support for a database server (DS). DB2 identifies a DS as a secondary server which can be a DRDA DS that services requests from a DB2 DRDA AS.

IBM DRDA supports DDM as follows:

- DRDA level 1 supports DDM level 3.
- DRDA level 2 supports DDM level 4.

Open Group DRDA supports DDM as follows:

- DRDA level 3 supports DDM level 5[1].
- DRDA level 4 supports DDM level 6[2].

Before using this book, you should have a good understanding of DRDA and Distributed Data Management (DDM).

Where to find more information

The DDM commands described in this book supplement the information in the following Open Group Technical Standard publications:

- *DRDA Volume 1: Distributed Relational Database Architecture (DRDA), ISBN 1-85912-295-7*
- *DRDA Volume 3: Distributed Database Management (DDM) Architecture, ISBN 1-85912-206-X*

The DDM Manager Level (MGRLVL) requirements for the DRDA levels are defined in *DRDA Volume 1: Distributed Relational Database Architecture (DRDA)*.

1 Open Group Technical Standard DRDA Version 1

2 Open Group Technical Standard DRDA Version 2

What product terminology and citations mean

In this book, DB2 Universal Database Server for OS/390 is referred to as "DB2 for OS/390." In cases where the context makes the meaning clear, DB2 for OS/390 is referred to as "DB2." When this book refers to other books in this library, a short title is used. (For example, "See *DB2 SQL Reference*" is a citation to *IBM DATABASE 2 Universal Database Server for OS/390 SQL Reference*.)

References in this book to "DB2 UDB" relate to the DB2 Universal Database™ product that is available on the AIX®, OS/2®, and Windows NT™ operating systems. When this book refers to books about the DB2 UDB product, the citation includes the complete title and order number.

The following terms are used as indicated:

DB2® Represents either the DB2 licensed program or a particular DB2 subsystem.

C and C language
Represent the C programming language.

CICS®
Represents CICS/ESA® and CICS Transaction Server for OS/390 Release 1.

IMS™ Represents IMS/ESA®.

MVS Represents the MVS/Enterprise Systems Architecture (MVS/ESA™) element of OS/390.

Summary of changes to this book

Changes to this edition of the book include:

- Support for large object (LOB) data types, see “Data type support” on page 4.
- A new command, EXCSQLSET, see “The EXCSQLSET command” on page 21.

Updates to commands are marked with revision bars.

DB2 distributed data management (DDM) command support

This chapter describes the DDM commands and command parameters, command data objects, and reply data objects that DB2 supports for DRDA level 1, DRDA level 2, and DRDA level 3.

An application requester (AR) using DRDA to connect to an application server (AS) uses a subset of *distributed data management* (DDM) as part of the underlying architecture of DRDA. DDM is the data connectivity language used for data interchange among like or unlike systems and is independent of a particular system's hardware architecture and its operating system.

Command data and reply data objects are defined by DDM; however, their structure is defined by the Formatted Data Object Content Architecture (FD:OCA).

Figure 1 illustrates the relationship between DDM and other architectures composing DRDA.

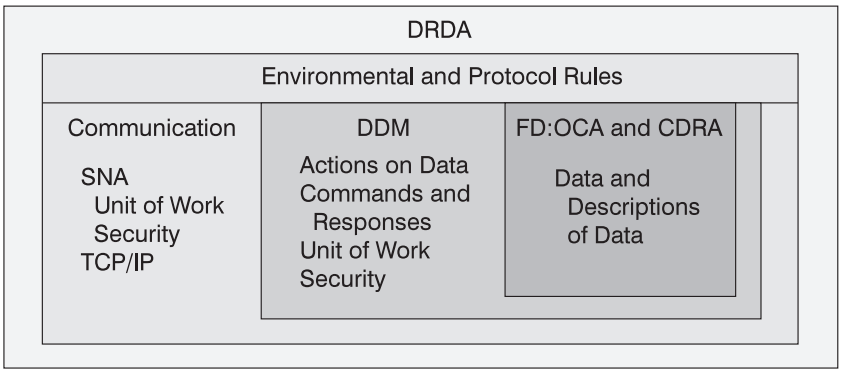


Figure 1. DDM's relationship to DRDA and other architectures. FD:OCA is the Formatted Data Object Content Architecture, and CDRA is the Character Data Representation Architecture.

DRDA is defined in terms of rules and usage of four underlying architectures:

- Communication in DRDA can use multiple network transport protocols. DB2 uses the following transport protocols:
  - SNA LU 6.2: Allows the specification of security requirements between the application and the relational database. Additional security requirements, such as OSF DCE (Open Software Foundation Distributed Computing Environment) that cannot be specified using SNA LU6.2, are specified using DDM commands and responses. SNA LU 6.2 also specifies the token (LUWID) that correlates the application process at the AR and the agent process at the AS. All commands and data are sent by way of SNA LU 6.2 conversations.
  - TCP/IP: All security and LUWID requirements are specified using DDM commands and responses. All commands and data are sent by way of TCP/IP sockets.
- Distributed Data Management (DDM) defines the syntax and semantics of all commands sent from an AR to an AS and all reply messages sent from an AS to an AR.

- Formatted Data Object Content Architecture describes the syntax and semantics of all command data objects sent from an AR to an AS and all reply data objects sent from an AS to an AR.
- Character Data Representation Architecture specifies the representation of character data sent within a command data object from an AR to an AS and character data sent within a reply data object from an AS to an AR.

Data type support

DRDA supports eight large object (LOB) data types. The following data types are fully supported and are accessible by any DB2 Version 6 application or by a DB2 Version 6 server when connected to a DRDA level 4 compliant server or requester:

- LOB bytes
- LOB SBCS character
- LOB DBCS character
- LOB MBCS character
- LOB bytes locator
- LOB character locator
- LOB DBCS character locator
- Row identifier

In addition, the following data types are supported using compatible data types:

- Eight-byte integer
- SBCS datalink
- MBCS datalink

If a DB2 application accesses an eight-byte integer on a DRDA server that fully supports eight-byte integers, the DB2 requester converts the description and values of the eight-byte integer to DECIMAL(19). If a DRDA requester provides an eight-byte integer from an input host variable, the DB2 server converts it to DECIMAL(19) before processing the SQL statement.

If a DB2 application accesses a datalink on a DRDA server that fully supports the datalink data type, the DB2 requester converts the description and values to a LONG VARCHAR. The application is required to parse out the contents of the datalink data type. See *DRDA Volume 1: Distributed Relational Database Architecture (DRDA)* for a complete description of the contents of a datalink data type. If a DRDA requester provides a datalink data type from an input host variable, the DB2 server converts it to a LONG VARCHAR before processing the SQL statement.

How to read the tables

A DDM command can have associated with it:

Parameters (known as instance variables in DDM)
Command data objects
Reply messages
Reply data objects.

These commands and their associated objects and messages map to actions taken by a relational database management system:

- SQL requests are sent to the AS as DDM commands.
- Application host variable descriptions and values are sent to the AS as DDM command data.
- The results of an SQL request are returned to the AR via DDM reply messages and DDM reply data objects.
- Descriptions of data returned to the application and the data itself are returned from the AS to the AR in DDM reply data objects.

If a particular command has any of the above objects associated with it, those objects are described in separate tables under the heading of that command.

DB2 supports all of the commands listed in this book.

For all the tables, the "**Required?**" column indicates whether the item is required by DRDA. The meanings of the "AR" and "AS" columns are described in the sections below.

Commands

The meanings of the columns for commands are:

The **AR** column indicates how the DB2 application requester supports the parameter:

Y means that DB2 sends it to the AS.

N means that DB2 does *not* send it to the AS.

I means that DB2 ignores it (when in a reply from the AS).

S means that DB2 supports a subset of DRDA options. The options are listed below the parameter in capital letters; Y means DB2 supports it and N means that DB2 does not send it.

The **AS** column indicates how the DB2 application server supports the code point or parameter:

Y means that DB2 recognizes and processes it.

N means that DB2 rejects it.

I means that DB2 ignores it if received.

S means that DB2 allows the parameter, depending on its value. The options are listed below the parameter in capital letters; Y means DB2 supports it and N means that DB2 rejects it.

Command data objects

The meanings of the columns for command data objects are:

The **AR** column indicates how the DB2 AR supports the parameter:

Y means that DB2 sends it to the AS.
N means that DB2 does *not* send it to the AS.

The **AS** column indicates how the DB2 AS supports the parameter:

Y means that DB2 receives and processes it.
I means that DB2 ignores the parameter.

Reply data objects

The meanings of the columns for reply data objects are:

The **AR** column indicates how the DB2 AR supports the parameter:

Y means that DB2 recognizes and processes it.
I means that DB2 ignores it.

The **AS** column indicates how the DB2 AS supports the parameter:

Y means that DB2 sends it to the AR.
N means that DB2 does *not* send it to the AR.

Reply messages

DDM reply messages fall into two classes. One class represents the reply messages returned in response to the normal processing of a DDM command. The other class contains reply messages returned in response to an error detected by the AS during the processing of a DDM command.

Normal situation

The normal message response is described under each DDM command that has a possible reply message. The meanings of the columns for reply messages are:

The **AR** column indicates how the DB2 AR supports the message:

Y means that DB2 always supports the receipt of any instance variable defined as valid for any DDM reply message.

N means that DB2 does *not* receive the message from the AS.

The **AS** column indicates how the DB2 AS supports the message:

Y means that DB2 sends it to the AR.
N means that DB2 does *not* send it to the AR.

Error situation

The DB2 AS can generate any of the following DDM reply messages if it fails to process a DDM command:

AGNPRMRM	MGRLVLRM	QRYPOPRM
BGNBNDRM	OBJNSPRM	RDBACCRM
CMDCHKRM	OPNQFLRM	RDBNACRM
CMDNSPRM	PKGBNARM	RSCLMTRM
CMDVLTRM	PKGBPARM	SQLERRRM
DSCINVRM	PRCCNVRM	SYNTAXRM
DTAMCHRM	PRMNSPRM	VALNSPRM
MGRDEPRM	QRYNOPRM	

The DB2 AS never sends any of the following DDM reply messages:

CMDATHRM	TRGNSPRM	RDBATHRM

RDBNFNRM is returned on:

- EXCSAT in releases prior to Version 5
- ACCSEC and SECCHK in Version 5

The ACCRDB command

The ACCRDB command means “access relational database,” and it establishes a path to a named relational database.

Table 1. ACCRDB command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database)	Y	Y	Y
Rdbacccl (access manager class)	Y	Y	Y
Typdefnam (data type definition name)	Y	Y	Y
Typdefovr (data type definition override)	Y	Y	Y
Prdid (product specific identifier)	Y	Y	Y
Rdbalwupd (rdb to allow updates)	Y	Y	N
Prddta (product specific data)	Y	Y	N
Note: See “Accounting for distributed data” on page 37 for more information.			
Sttstrdel (string delimiter) STRDELAP STRDELDQ DFTPKG	Y Y Y N	Y Y Y Y	N
Sttdecdel (decimal delimiter) DECDELPRD DECDELCMA DFTPKG	S Y Y N	Y Y Y Y	N
Crrtkn (correlation token)	Y	Y	N
Trgdftrt (target default values return)	N	Y	N

There are no command data objects or reply objects as defined by DRDA for ACCRDB.

Table 2. ACCRDBRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Prdid (product identifier)	Y	Y	Y
Typdefnam (data type definition name)	Y	Y	Y
Rdbinttkn (RDB interrupt token)	Y	N	N
Crrtkn (correlation token)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N
Pkgdftcst (package default character subtype)	Y	N	N
Usrid (user ID at the target system)	Y	N	N
Srvlst (server list)*	Y	Y	N

Note: * Srvlst is supported by DRDA level 3 only.

The ACCSEC command

The ACCSEC command means “access security.” It initializes the security mechanism used to validate the users identity.

Table 3. ACCSEC command instance variables

Instance variable	AR	AS	Required?
Secmgrnm (security manager name)	N	I	N
Secmec (security mechanism)	Y	Y	Y
Rdbnam (target RDB)	Y	Y	N
Sectkn (security token)	N	Y	N

There are no command data objects defined by DRDA for ACCSEC.

ACCSECRD reply object for ACCSEC command

There is only one reply data object defined as valid for ACCSEC. This is the ACCSECRD reply data object. This table lists instance variables for this object, because some are optional and not supported as a requester.

Table 4. ACCSECRD reply object instance variables

Instance variable	AR	AS	Required?
Sectkn (security token)	N	Y	N
Secmec (security mechanism)	Y	Y	Y
Secchkcd (security check code)	N	Y	N

There are no DDM reply messages returned as a result of normal processing of the ACCSEC command.

The BGNBND command

The BGNBND command means “begin bind,” and it starts the process of binding a package into a particular relational database.

Table 5 (Page 1 of 2). BGNBND command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamct (package name and consistency token)	Y	Y	Y
Vrsnam (package version name)	Y	Y	N
Bndchkexs (bind existence checking)	Y	Y	N
Pkgrplopt (package replacement option)	Y	Y	N
Pkgathopt (package authorization option)	Y	Y	N
Pkgathrul (package authorization rules)	Y	Y	N
OWNER	Y	Y	
REQUESTER	Y	Y	
DEFINER_REVERT_TO_REQUESTER*	Y	Y	
DEFINER_REVERT_TO_OWNER*	Y	Y	
INVOKER_REVERT_TO_REQUESTER*	Y	Y	
INVOKER_REVERT_TO_OWNER*	Y	Y	
Sttstrdel (statement string delimiter)	Y	Y	N
Sttdecdel (statement decimal delimiter)	Y	Y	N
Sttdatfmt (date format of statement)	Y	Y	N
Stttimfmt (time format of statement)	Y	Y	N
Pkgisolvl (package isolation level)	S	Y	Y
ISOLVLRR	Y	Y	
ISOLVLALL	Y	Y	
ISOLVLCS	Y	Y	
ISOLVLCHG	Y	Y	
ISOLVLNC	Y	Y	
Dgrioprl (degree of I/O parallelism)	Y	Y	N
Bndcrtctl (bind creation control)	S	Y	N
BNDCHKONL	N	Y	
BNDERRALW	Y	Y	
BNDNERALW	Y	Y	
Bndexpopt (bind explain option)	Y	Y	N
Pkgownid (package owner identifier)	Y	Y	N
Rdbrlsopt (RDB release option)	Y	Y	N
Dftrdbcol (default RDB collection identifier)	Y	Y	N
Title (brief description of package)	N	I	N
Qryblkctl (query block protocol control)	Y	Y	N
Pkgdftcst (default character subtype)	N	S	N
CSTSYDFT	N	Y	
CSCTBITS	N	I	
CSTSBCS	N	I	
CSTMBCS	N	I	
Pkgdftcc (package default CCSID)	N	Y	N

Table 5 (Page 2 of 2). BGNBND command instance variables

Instance variable	AR	AS	Required?
Pkgrplvrs (replaced package version name)	Y	Y	Y
Decprc (decimal precision)	N	Y	N

Note:

***** DEFINER_REVERT_TO_REQUESTER, DEFINER_REVERT_TO_OWNER, INVOKER_REVERT_TO_REQUESTER, and INVOKER_REVERT_TO_OWNER are supported by DRDA level 4 and above only.

Table 6. BGNBND command data objects

Command data object	AR	AS	Required?
Bndopt (bind option)*	Y	Y	N

Note: Bndopt is supported by DRDA level 3 only.

Table 7. Reply objects for BGNBND command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 8. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The BNDSQLSTT command

The BNDSQLSTT command means “bind SQL statement,” and it binds an SQL statement to a package.

Table 9. BNDSQLSTT command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
Sqlsttnbr (source application statement number)	Y	Y	N
Bndsttasm (bind statement assumptions)	Y	Y	N

Table 10. BNDSQLSTT command data objects

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlstt (SQL statement to be bound in the AS package)	Y	Y	Y
Sqlsttvrb (description of each variable)	Y	Y	N

Table 11. Reply objects for BNDSQLSTT command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 12. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The CLSQRY command

The CLSQRY command means “close query,” and it terminates a query. It corresponds to a CLOSE.

Table 13. CLSQRY command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn	Y	Y	Y

There are no command data objects defined by DRDA for CLSQRY.

Table 14. Reply objects for CLSQRY command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

There are no DDM reply messages returned as a result of normal processing of the CLSQRY command.

The CNTQRY command

The CNTQRY command means “continue query,” and it is a request to resume a query that was interrupted.

Table 15. CNTQRY command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
Qryblksz (query block size)	Y	Y	Y
Maxblkext (maximum number of additional query blocks)*	Y	Y	N
Rtnextdta (return of EXTDTA option)*	S	Y	N
RTNEXTROW (return EXTDTAs by row)	Y	Y	
RTNEXTALL (return all EXTDTAs for QRYDTA sent)	N	Y	

Note: * Maxblkext and Rtnextdta are supported by DRDA level 4 and above only.

Table 16. CNTQRY command data objects

Reply object	AR	AS	Required?
Outovr (output override descriptor)	Y	Y	N

Table 17. Reply objects for CNTQRY command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Qrydta (query answer set data)	Y	Y	N
Extdta (externalized FD:OCA data) *	Y	Y	N

Note: * Extdta is supported by DRDA level 4 and above only.

Table 18. ENDQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	N	N

The DRPPKG command

The DRPPKG means “drop package,” and it deletes a named package from a relational database instance.

Table 19. DRPPKG command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnam (package grouping name and identifier)	Y	Y	Y
Vrsnam (version name)	Y	Y	N

There are no command data objects defined by DRDA for DRPPKG.

Table 20. Reply objects for DRPPKG command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 21. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The DSCRDBTBL command

The DSCRDBTBL command means “describe table,” and it is a request for a description of the RDB table named in the SQLTBLNAM command data object being returned to the requester.

Table 22. DSCRDBTBL command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N

Table 23. DSCRDBTBL command data objects

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
Sqltblnam (SQL table name)	Y	Y	Y

Table 24. Reply objects for DSCRDBTBL command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Sqldard (SQLDA reply data)	Y	Y	Y

There are no DDM reply messages returned as a result of normal processing of the DSCRDBTBL command.

The DSCSQLSTT command

The DSCSQLSTT command means “describe SQL statement,” and it is a request for definitions of the columns of the result table of a prepared or bound statement and the names and labels of those columns. DB2 sends a DSCSQLSTT command only when executing the SQL DESCRIBE INPUT statement.

Table 25. DSCSQLSTT command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
TYPSQLDA (type of SQL descriptor area,	S	Y	N
input or output)		Y	
0 (return output SQLDA, default)	N	Y	
1 (return input SQLDA)	Y		

There are no command data objects defined by DRDA for DSCSQLSTT.

Table 26. Reply objects for DSCSQLSTT command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Sqldard (SQLDA reply data)	Y	Y	Y

There are no DDM reply messages returned as a result of normal processing of the DSCSQLSTT command.

The ENDBND command

The ENDBND command means “end bind.” It indicates that no more BIND commands will be sent, and the package is now complete.

Table 27. ENDBND command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamct (package name and consistency token)	Y	Y	Y
Maxsctnbr (maximum section number)	Y	Y	N

There are no command data objects defined by DRDA for ENDBND.

Table 28. Reply objects for ENDBND command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 29. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The EXCSAT command

The EXCSAT means “exchange server attributes,” and it is the first DDM command sent from a requester to a server. The requester communicates the minimum level of support which it requires from the server.

Table 30. EXCSAT command instance variables

Instance variable	AR	AS	Required?
Extnam (external name)	Y	Y	N
Mgrlvlls (manager level list)	Y	Y	N
Spvnam (supervisor name)	N	N	N
Srvclsnm (server class name)	Y	Y	N
Srvnam (server name)	Y	Y	N
Srvrlslv (server release level)	Y	Y	N

There are no command data objects defined by DRDA for EXCSAT.

EXCSATRD reply object for the EXCSAT command

There is only one reply data object defined as valid for EXCSAT. This is the EXCSATRD reply data object. This table lists instance variables for this object, because they are optional and can vary.

Table 31. EXCSATRD reply object instance variables

Instance variable	AR	AS	Required?
Extnam (external name)	Y	Y	N
Mgrlvlls (manager level list)	Y	Y	N
Srvclsnm (server class name)	Y	Y	N
Srvnam (server name)	Y	Y	N
Srvrlslv (server release level)	Y	Y	N

There are no DDM reply messages returned as a result of normal processing of the EXCSAT command.

The EXCSQLIMM command

The EXCSQLIMM command means “execute SQL statement immediate,” and it executes the single SQL statement sent with the command.

Table 32. EXCSQLIMM command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y

Table 33. Command data objects for EXCSQLIMM command

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
Sqlstt (SQL statement)	Y	Y	Y

Table 34. Reply objects for EXCSQLIMM command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 35. ENDUOWRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Uowdsp (unit of work disposition)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

Table 36. CMMRQSRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Cmmtyp (commitment request type)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

Table 37. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The EXCSQLSET command

The EXCSQLSET command means "execute SQL set," and it executes one or more SET statements to establish the application environment.

Table 38. EXCSQLSET command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	Y
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y

Table 39. Command data objects for EXCSQLSET command

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
SQLSTT (SQL statement)	Y	Y	Y

Note: See “Accounting for distributed data” on page 37 for more information.

Table 40. Reply objects for EXCSQLSET command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

There are no DDM reply messages returned as a result of normal processing of the EXCSQLSET command.

The EXCSQLSTT command

The EXCSQLSTT command means “execute SQL statement,” and it executes a previously bound SQL statement.

Table 41. EXCSQLSTT command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Prcnam (stored procedure name)	Y	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
Outexp (output expected)	Y	Y	N
Maxrslcnt (maximum number of result sets)(1) -1	S Y	Y Y	N
Maxblkext (maximum number of additional query blocks)(2)	Y	Y	N
Qryblksz (query block size)	N	Y	Y
Rslsetflg (result set flag)	Y	Y	N
Rdbcmtok (commit by AS is allowed)(3)	N	Y	N
Outovropt (output override option)(2) OUTOVRFRS (output override allowed on on the first CNTQRY) OUTOVRANY (output override allowed on any CNTQRY)	Y Y Y	Y Y Y	N

Notes:

(1) Maxrslcnt is supported by DRDA level 3 only.
(2) Maxblkext and Outovropt are supported by DRDA level 4 and above only.
(3) Rdbcmtok is supported by DRDA level 2 only.

Table 42. Command data objects for the EXCSQLSTT command

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
Sqldta (SQL program variable data)	Y	Y	N
Prcnam (stored procedure name)	Y	Y	N
Extdta (externalized FD:OCA data)	Y	Y	N
Outovr (output override descriptor) *	Y	Y	N

Note: * Outovr is supported by DRDA level 4 and above only.

Table 43 (Page 1 of 2). Reply objects for EXCSQLSTT command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Sqldtard (SQL data reply data)	Y	Y	N

Table 43 (Page 2 of 2). Reply objects for EXCSQLSTT command

Reply object	AR	AS	Required?
Sqlcinrd (SQL column information reply data)	Y	Y	N
Sqlrslrd (query result set returned by stored procedure)(1)	Y	Y	N
Qrydsc (query answer set description)(1)	Y	Y	N
Qrydta (query answer set data)(1)	Y	Y	N
Extdta (externalized FD:OCA data)(2)	Y	Y	N

Notes:

(1) Sqlrslrd, Qrydsc, and Qrydta are supported by DRDA level 3 only.
(2) Extdta is supported by DRDA level 4 or above only.

Table 44. ENDUOWRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Uowdsp (unit of work disposition)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

Table 45. CMMRQSRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Cmmtyp (commitment request type)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

Table 46. RDBUPDRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

Table 47. RSLSETRM reply message instance variables for DRDA level 3 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Pkgsnlst (package section list)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

Table 48 (Page 1 of 2). OPNQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Qryprctyp (query protocol type)	Y	Y	Y

Table 48 (Page 2 of 2). OPNQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Sqlcsrhld (hold cursor position)	Y	Y	N
Srvdgn (server diagnostic information)	Y	N	N

Table 49. ENDQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	N	N

The OPNQRY command

The OPNQRY command means “open query,” and it is a request to begin the query process. It is sent as a result of an SQL OPEN statement.

Table 50. OPNQRY command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
Qryblksz (query block size)	Y	Y	Y
Qryblkctl (query block protocol control)	N	Y	N
Maxblkext (maximum number of additional query blocks)*	Y	Y	N
Outovropt (output override option) *	Y	Y	N
OUTOVRFRS (output override allowed on on the first CNTQRY)	Y	Y	
OUTOVRANY (output override allowed on any CNTQRY)	Y	Y	

Note: * Maxblkext and Outovropt are supported by DRDA level 4 and above only.

Table 51. Command data objects for OPNQRY command

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
Sqldta (input variable data)	Y	Y	N
Extdta (externalized FD:OCA data)*	Y	Y	N

Note: * Extdta is supported by DRDA level 4 and above only.

Table 52. Reply objects for OPNQRY command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Qrydsc (query answer set description)	Y	Y	Y
Qrydta (query answer set data)	Y	Y	N

Table 53. OPNQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Qryprctyp (query protocol type)	Y	Y	Y
Sqlcsrhld (hold cursor position)	Y	Y	N
Srvdgn (server diagnostic information)	Y	N	N

Table 54. ENDQRYRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	N	N

| See “The PRPSQLSTT command” on page 27 for information about a resource
| limit condition that can exist when the OPNQRY command is chained to the
| PRPSQLSTT command.

The PRPSQLSTT command

The PRPSQLSTT means “prepare an SQL statement,” and it dynamically binds a single SQL statement to a section number in an existing package in a relational database instance.

Table 55. PRPSQLSTT command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnamcsn (package name, consistency token and section number)	Y	Y	Y
Rtnsqlda (specifies if SQLDA should be returned)	Y	Y	N

Table 56. Command data objects for PRPSQLSTT command

Command data object	AR	AS	Required?
Typdefnam (data type definition name)	N	Y	N
Typdefovr (data type definition override)	N	Y	N
Sqlstt (SQL Statement)	Y	Y	Y

Table 57. Reply objects for PRPSQLSTT command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	N
Sqldard (SQLDA reply data)	Y	Y	N

There are no DDM reply messages returned as a result of normal processing of the PRPSQLSTT command.

If the DB2 for OS/390 predictive governing function is active, a PRPSQLSTT command that is sent to a DB2 AS can receive a SQLCARD or SQLDARD with warning SQLSTATE 01616 (SQLCODE +495). This occurs due to a resource limit condition that exists if the prepared section is used in a subsequent OPNQRY command. If an OPNQRY command is chained to a PRPSQLSTT command that receives warning SQLSTATE 01616, the chained OPNQRY receives an SQLCARD with error SQLSTATE 57057 (SQLCODE -30002). An AR that detects SQLSTATE 57057 can take one of the following actions:

- Send the OPNQRY command to the AS again to perform the open. A DB2 AR takes this action.
- Send other commands to the AS, deferring or omitting the OPNQRY command.
- Take a conditional action, such as prompting the end user, to determine whether to send the OPNQRY command to the AS again. An interactive DRDA application can use this approach.

For additional DB2 information, see the descriptions of SQLCODEs +495 and -30002 in *DB2 Messages and Codes*. For additional DRDA information, see the

description of passing warnings to the AR (WN rules) in *DRDA Volume 1: Distributed Relational Database Architecture (DRDA)*.

The RDBCMM command

The RDBCMM command means “commit transaction,” and it commits the current unit of work.

Table 58. RDBCMM command instance variable

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	Y	Y	N

There are no command data objects defined by DRDA for RDBCMM.

Table 59. Reply objects for RDBCMM command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 60. ENDUOWRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Uowdsp (unit of work disposition)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

Table 61. CMDVLTRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The RDBRLLBCK command

The RDBRLLBCK command means, “roll back transaction,” and it rolls back (backs out) the current unit of work.

Table 62. RDBRLLBCK command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	Y	Y	N

There are no command data objects defined by DRDA for RDBRLLBCK.

Table 63. Reply objects for RDBRLLBCK command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 64. ENDUOWRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Uowdsp (unit of work disposition)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

Table 65. CMDVLTRM reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The REBIND command

The REBIND command results in the rebind of an existing package at the server. No SQL statements are sent with REBIND. The SQL statements must have been bound previously at the server.

Table 66. REBIND command instance variables

Instance variable	AR	AS	Required?
Rdbnam (name of remote database as in ACCRDB)	N	Y	N
Pkgnam (package name)	Y	Y	Y
Vrsnam (package version name)	Y	Y	N
Pkgathrul (package authorization rules)	Y	Y	N
OWNER	Y	Y	
REQUESTER	Y	Y	
DEFINER_REVERT_TO_REQUESTER*	Y	Y	
DEFINER_REVERT_TO_OWNER*	Y	Y	
INVOKER_REVERT_TO_REQUESTER*	Y	Y	
INVOKER_REVERT_TO_OWNER*	Y	Y	
Pkgisolvl (package isolation level)	S	Y	Y
ISOLVLRR	Y	Y	
ISOLVLALL	Y	Y	
ISOLVLCS	Y	Y	
ISOLVLCHG	Y	Y	
ISOLVLNC	Y	Y	
Bndexpopt (bind explain option)	Y	Y	N
Pkgownid (package owner identification)	Y	Y	N
Rdbrlsopt (RDB release option)	Y	Y	N
Bndchkexs (bind existence checking)	Y	Y	N
Dftrdbcol (default RDB collection identifier)	Y	Y	N
Dgrioprl (degree of I/O parallelism)	Y	Y	N

Note:

* DEFINER_REVERT_TO_REQUESTER, DEFINER_REVERT_TO_OWNER, INVOKER_REVERT_TO_REQUESTER, and INVOKER_REVERT_TO_OWNER are supported by DRDA level 4 and above only.

Table 67. REBIND command data objects

Command data object	AR	AS	Required?
Bndopt (bind option)	Y	Y	N

Table 68. Reply objects for REBIND command

Reply object	AR	AS	Required?
Typdefnam (data type definition name)	Y	N	N
Typdefovr (data type definition override)	Y	N	N
Sqlcard (SQLCA reply data)	Y	Y	Y

Table 69. RDBUPDRM Reply message instance variables for DRDA level 2 only

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Rdbnam (relational database name)	Y	Y	Y
Srvdgn (server diagnostic information)	Y	Y	N

The SECCHK command

The SECCHKRM command means “security check.” It sends information to the target security manager to authenticate the user.

Table 70. SECCHK command instance variables

Instance variable	AR	AS	Required?
Secmgrnm (security manager name)	Y	Y	N
Secmec (security mechanism)	Y	Y	Y
Password (password)	Y	Y	N
Usrid (user ID at the target system)	Y	Y	N
Rdbnam (target RDB)	Y	Y	N
Sectkn (security token)	N	Y	N

Table 71. SECCHK command data objects

Command data object	AR	AS	Required?
Sectkn (security token)	Y	Y	N

Table 72. Reply objects for SECCHK command

Reply object	AR	AS	Required?
Sectkn (security token)	Y	Y	N

Table 73. SECCHKRM reply message instance variables

Instance variable	AR	AS	Required?
Svrcod (severity code)	Y	Y	Y
Secchkcd (security check code)	Y	Y	Y
Svcerrno (security service error number)	Y	Y	N
Srvdgn (server diagnostic information)	Y	Y	N

The SYNCCTL command

The SYNCCTL command means “sync point control.” It conveys sync point information to the target.

Table 74. SYNCCTL command instance variables

Instance variable	AR	AS	Required?
Synctype (sync point operation type)	Y	Y	Y
Rlsconv (release conversation)	Y	Y	N
Uowid (unit of work identifier)	Y	Y	N
Forget (forget reply required)	Y	Y	N

Table 75. SYNCLOG data object for SYNCCTL command

Data Object	AR	AS	Required?
Rdbnam (RDB associated with log)	Y	Y	Y
Lognam (log name)	Y	Y	Y
Logtstmp (log time stamp)	Y	Y	Y
Snaaddr (resync SNA address)	Y	Y	N
IPaddr (resync TCP/IP address)	Y	Y	N
Tcphost (fully qualified host domain name)	Y	Y	N
Cnntkn (connection token)	Y	Y	Y

Table 76. SYNCCRD reply object for SYNCCTL command

Reply object	AR	AS	Required?
Synctype (sync point operation type)	Y	Y	Y
Synclog (sync point log)	Y	Y	N

There are no DDM reply messages returned as a result of normal processing of the SYNCCTL command.

The SYNCRSY command

The SYNCRSY command means “sync point resync.” It resolves indoubt units of work between sync point managers.

Table 77. SYNCRSY command instance variables

Instance variable	AR	AS	Required?
Rsynctyp (resync operation type)	Y	Y	Y
Uowid (unit of work identifier)	Y	Y	N
Uowstate (unit of work state)	Y	Y	N

Table 78. Command data objects for SYNCRSY command

Command data object	AR	AS	Required?
Synclog (sync point log)	Y	Y	N

Table 79. SYNCRRD reply object for SYNCRSY command

Reply object	AR	AS	Required?
Rsynctyp (resync operation type)	Y	Y	Y
Uowid (UOW identifier)	Y	Y	Y
Uowstate (UOW state)	Y	Y	Y

There are no DDM reply messages returned as a result of normal processing of the SYNCRSY command.

Accounting for distributed data

To enable an accounting system or a monitoring system to track DRDA access to a DB2 application server (AS), the application requester (AR) can send accounting and monitoring data to DB2. There are two ways to send the data:

- Send an accounting identifier string in the PRDDTA instance variable of the ACCRDB command to the DB2 AS with each application's connect request. The format of the accounting string in PRDDTA is described below.
- Send accounting or monitoring identifier strings in the SQLSTT command data object of the EXCSQLSET command to the DB2 AS. This is not limited to connect requests. The format of the accounting and monitoring string information in SQLSTT is described in “Format of SQLSTT” on page 40.

Macro DSNDQMDA maps the accounting record. For a detailed description of the fields in this record, refer to this mapping macro in the data set library DSN610.SDSNMACS.

Format of PRDDTA

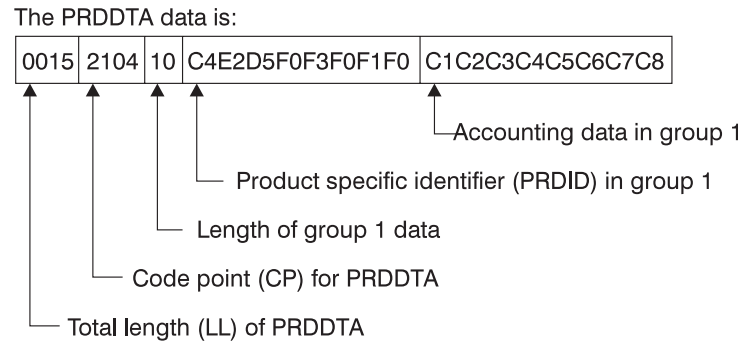
The data in the PRDDTA instance variable must be in one or more repeating groups of:

L One-byte length of the data that follows. The length can be zero if no data follows for this group.

Data Data for length L.

If the length of all repeating groups equals the total length of PRDDTA, DB2 assumes that the **first** group is accounting data.

Figure 2 on page 38 shows an example of one group and Figure 3 on page 39 shows an example of four repeating groups. All values are expressed in hexadecimal.



The length of the PRDDTA is calculated by adding all the parts:

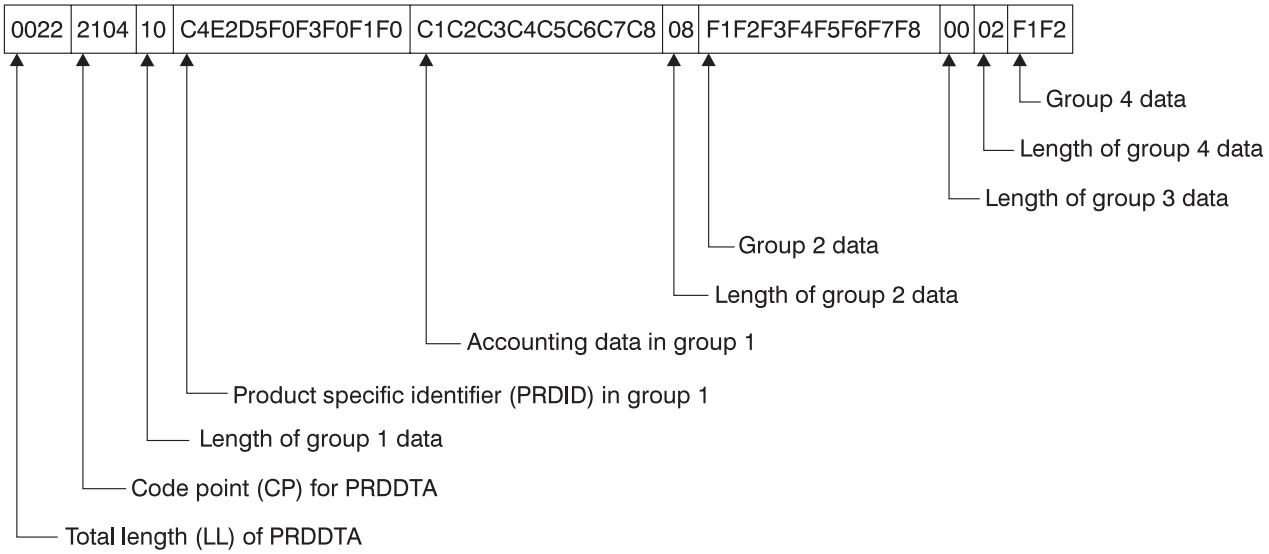
02 Bytes of LL
02 Bytes of CP
01 Byte length for group 1
10 Bytes of data in group 1

15 Total length of PRDDTA

Because the sum of the parts equals LL, DB2 assumes that the first group (the only group in this case) contains accounting data.

Figure 2. Example of one group in PRDDATA

The PRDDTA data is:



The length of the PRDDTA is calculated by adding all the parts:

02 Bytes of LL
02 Bytes of CP
01 Byte length for group 1
10 Bytes of data in group 1
01 Byte length for group 2
08 Bytes of data in group 2
01 Byte length for group 3
00 Bytes of data in group 3
01 Byte length for group 4
02 Bytes of data in group 4

22 Total length of PRDDTA

Because the sum of the parts equals LL, DB2 assumes that the first group contains accounting data.

Figure 3. Example of four repeating groups in PRDDATA

Format of SQLSTT

DB2 server systems allow requester systems to influence certain accounting and monitoring information using the EXCSQLSET command. DB2 server systems have implemented the concepts of:

- End user user IDs
- End user workstation names
- End user application names
- Accounting data

The PRDDTA instance variable can also influence this information. However, PRDDTA only allows the accounting data to be sent when the connection is first established, whereas the EXCSQLSET command allows the accounting data to be sent at any time.

Much of this information is externalized in various forms in a DB2 system. For example:

- The DSNV437I message of the DISPLAY THREAD command report
- THREAD-INFO data in various messages such as DSNT375I
- The QWHC trace record correlation header
- The QMDA section of DB2 accounting trace records

The EXCSQLSET command is sent in conjunction with one or more SQLSTT command data objects. The format of the SQLSTT determines the information to be sent as follows:

- End user user ID

  To set the end user user ID, SQLSTT should contain the string `SET CLIENT USERID` followed by the user ID in single quotes. DB2 accepts a user ID of up to 16 characters and truncates any characters that exceed that length. For example:

  ```
  SET CLIENT USERID 'my_eu_userid'
  ```

- End user workstation name

  To set the end user workstation name, SQLSTT should contain the string `SET CLIENT WRKSTNNAME` followed by the workstation name in single quotes. DB2 accepts a name of up to 18 characters and truncates any characters that exceed that length. For example:

  ```
  SET CLIENT WRKSTNNAME 'my_eu_wstationname'
  ```

- End user application name

  To set the end user application name, SQLSTT should contain the string `SET CLIENT APPLNAME` followed by the application name in single quotes. DB2 accepts a name of up to 32 characters and truncates any characters that exceed that length. For example:

  ```
  SET CLIENT APPLNAME 'my_eu_applname'
  ```

- Accounting information

  To set the accounting information, SQLSTT should contain the string `SET CLIENT ACCTNG` followed by the accounting information in single quotes. DB2 accepts up to 255 characters and truncates any characters that exceed that

length. DB2 also assumes that the first 8 characters of accounting information are a product identifier (PRDID). For example:

```
SET CLIENT ACCTNG 'DSN05010my_acctng_info'
```

A comma (,) is used to delimit string information. Hexadecimal string values can be represented by `X'hh'`. For example:

```
SET CLIENT ACCTNG 'DSN05010my_accntg_info',X'0004','ABCD'
SET CLIENT ACCTNG 'DSN05010','my_accntg_info',X'0004','ABCD'
SET CLIENT ACCTNG 'DSN05010','my_accntg_info',X'00',X'04','AB','CD'
```

Distributing work in a data sharing group

DB2 uses the Sysplex transaction program, an LU6.2 TPN (Transaction Program Name), in a SNA environment to allow DRDA requesters and DB2 private protocol requesters to determine which DB2 servers are currently active within the DB2 data sharing group. The TPN returns a list of DB2 server NETID.LUNAME values. The server list (Srvlst) instance variable on ACCRDB provides a list of DB2 server IPaddr.PORT values that are weighted based on the available capacity at each DB2 server in a TCP/IP environment. This allows the requester to distribute future requests for DB2 threads across the available DB2 servers in the group on the basis of capacity.

The LU6.2 parameters and messages for invoking this transaction are described here.

Requirements: The Sysplex transaction program requires MVS/ESA Version 5 Release 2 or subsequent releases.

Allocating a conversation to the Sysplex transaction program

The parameters used to allocate an LU6.2 conversation to the Sysplex transaction program are:

- TPN — X'03F0F3C2' (TPN prefix X'03'' and TPN suffix C'03B').
- SECURITY(NONE)
- SYNC_LEVEL(NONE)
- CONVERSATION_TYPE(BASIC)

Format of input message

The format of the Sysplex transaction program input message is:

Offset	Length	Description
0	2	Length of the input message, including the 2-byte length field. The value of this field is 4 (X'0004').
2	2	Type of input message. The only defined value at this time is X'F0F0', which indicates the requester wants to receive the NETID.LUNAMEs of the DB2 subsystems in the DB2 group.

Format of reply message

The format of the DB2 Sysplex transaction program reply message is:

Offset	Length	Description
0	2	Length of the reply message, including the 2-byte length field. The value of this field is 22+(*n*×18), where *n* is the number of members in the data sharing group.
2	2	Type of reply message. The only defined value at this time is X'F0F1', which indicates this is a reply message containing a list of DB2 server NETID.LUNAME values.
4	18	The location name of the DB2 server. This value is returned by the server as a verification aid to the requester. In this way, the requester can detect cases where the Sysplex transaction program was directed to the wrong NETID.LUNAME because of errors in the communication directory or CDB entries.
22	n ×18	*n* occurrences of the following: **NETID** VTAM® network name of the DB2 server group. This is 8 characters, padded on the right with blanks. **LUNAME** VTAM LU name of the DB2 server group. This is 8 characters, padded on the right with blanks. **Weight** A 2-byte integer containing the weighting factor for the server identified by NETID.LUNAME. This number controls the proportion of LU6.2 conversations directed to the server identified by NETID.LUNAME. The value X'FFFF' is reserved, indicating that the server is not part of a data sharing group. How the weighting factor works: Assume the value of *n* is 2. Weight 1 is 4. Weight 2 is 1. With these values, 80% of the LU6.2 conversations should be directed to the first NETID.LUNAME and 20% should be directed to the second NETID.LUNAME. The entries in the list are ordered by the weighting factor, with the greatest weight listed first. The value of *n* is always less than or equal to 32.

Appendix A. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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.

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
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
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.

This information contains 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 contains 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.

Programming Interface Information

This book is intended to help you to write programs that communicate with IBM DATABASE 2 Server for OS/390 (DB2 for OS/390) by means of the commands of distributed data management (DDM).

This book documents General-use Programming Interface and Associated Guidance Information provided by DB2 for OS/390.

General-use programming interfaces allow the customer to write programs that obtain the services of DB2 for OS/390.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

AIX
CICS
CICS/ESA
DATABASE 2
DB2

DB2 Universal Database
Distributed Relational Database Architecture
DRDA
IBM
IMS
IMS/ESA
MVS/ESA
OS/2
OS/390
VTAM

Microsoft™, Windows™, Windows NT, and the Windows logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

How to send your comments

DB2 Universal Database for OS/390
Reference for Remote DRDA
Requesters and Servers
Version 6

Publication No. SC26-9012-00

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 for OS/390 documentation. You can use any of the following methods to provide comments.

- Send your comments from the Web. Visit the DB2 for OS/390 Web site at:

  http://www.software.ibm.com/data/db2/os390

  The Web site has a feedback page that you can use to send comments.
- Send your comments by e-mail to db2pubs@vnet.ibm.com and include the name of the product, the version number of the product the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title, page number, or a help topic title).
- Complete the readers' comment form at the back of the book and return it by mail, by fax (800-426-7773 for the United States and Canada), or by giving it to an IBM representative.

Readers' Comments

DB2 Universal Database for OS/390
Reference for Remote DRDA
Requesters and Servers
Version 6

Publication No. SC26-9012-00

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐
Grammatically correct and consistent	☐	☐	☐	☐	☐
Graphically well designed	☐	☐	☐	☐	☐
Overall satisfaction	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

May we contact you to discuss your comments? ☐ Yes ☐ No

Name

Address

Company or Organization

Phone No.

Readers' Comments
SC26-9012-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department BWE/H3
PO Box 49023
San Jose, CA 95161-9945

Fold and Tape | **Please do not staple** | Fold and Tape

SC26-9012-00

Cut or Fold
Along Line

Program Number: 5645-DB2

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-9012-00