



**Reference for Remote DRDA  
Requesters and Servers**





**Reference for Remote DRDA  
Requesters and Servers**

**Note**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 61.

**First Edition (March 2004)**

This edition applies to Version 8 of IBM DB2 Universal Database for z/OS (DB2 UDB for z/OS), product number 5625-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.

© Copyright International Business Machines Corporation 1992, 2004. All rights reserved.

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

---

# Contents

<b>About this book</b> . . . . .	v
Who should read this book . . . . .	v
Terminology and citations . . . . .	v
Accessibility . . . . .	vi
How to send your comments . . . . .	vi
<b>Summary of changes to this book</b> . . . . .	vii
<b>Chapter 1. DDM command support</b> . . . . .	1
DRDA architecture . . . . .	2
Data type support . . . . .	3
Bookmarks with scrollable cursor support . . . . .	4
Query processing rules . . . . .	6
Monitoring . . . . .	6
How to read the DDM command tables . . . . .	6
ACCRDB command . . . . .	10
ACCSEC command . . . . .	12
BGNBND command . . . . .	14
BNDSQLSTT command . . . . .	17
CLSQRY command . . . . .	19
CNTQRY command . . . . .	20
DRPPKG command . . . . .	22
DSCRDBTBL command . . . . .	23
DSCSQLSTT command . . . . .	24
ENDBND command . . . . .	25
EXCSAT command . . . . .	26
EXCSQLGET command . . . . .	27
EXCSQLIMM command . . . . .	28
EXCSQLSET command . . . . .	30
EXCSQLSTT command. . . . .	31
INTRDBRQS command. . . . .	35
OPNQRY command . . . . .	36
PRPSQLSTT command. . . . .	39
RDBCMM command . . . . .	41
RDBRLLBCK command . . . . .	43
REBIND command . . . . .	45
SECCHK command . . . . .	47
SNDPKT command . . . . .	48
SYNCCTL command. . . . .	49
SYNCRSY command . . . . .	51
<b>Chapter 2. Accounting for distributed data.</b> . . . . .	53
Format of prddta data . . . . .	53
Format of sqlstt data . . . . .	56
<b>Chapter 3. Distributing work across data sharing members in an SNA network</b> . . . . .	59
Allocating a conversation to the Sysplex transaction program . . . . .	59
Format of input message . . . . .	59
Format of reply message . . . . .	60
<b>Notices</b> . . . . .	61
Programming interface information. . . . .	62

Trademarks . . . . . 62

---

## About this book

This book describes:

- The Distributed Data Management (DDM) commands that DB2 supports
- How an accounting system can track Distributed Relational Database Architecture™ (DRDA®) access to DB2
- How an application requester can distribute requests among the members of a data sharing group

---

## Who should read this book

This book is intended for anyone who is implementing a requester or a database server that communicates with DB2 using DRDA.

Before using this book, you should have a good understanding of DRDA and DDM.

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

- *Open Group Technical Standard, DRDA Version 3 Vol. 1: Distributed Relational Database Architecture*
- *Open Group Technical Standard, DRDA Version 3 Vol. 3: Distributed Data Management Architecture*

The DDM Manager Level (MGRLVL) requirements for the DRDA levels are defined in *Open Group Technical Standard, DRDA Version 3 Vol. 1: Distributed Relational Database Architecture*.

---

## Terminology and citations

In this information, DB2 Universal Database™ for z/OS™ is referred to as "DB2 UDB for z/OS." In cases where the context makes the meaning clear, DB2 UDB for z/OS is referred to as "DB2®." When this information refers to titles of books in this library, a short title is used. (For example, "See *DB2 SQL Reference*" is a citation to *IBM® DB2 Universal Database for z/OS SQL Reference*.)

When referring to a DB2 product other than DB2 UDB for z/OS, this information uses the product's full name to avoid ambiguity.

The following terms are used as indicated:

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

**DB2 PM**

Refers to the DB2 Performance Monitor tool, which can be used on its own or as part of the DB2 Performance Expert for z/OS product.

**C, C++, and C language**

Represent the C or C++ programming language.

**CICS®** Represents CICS Transaction Server for z/OS or CICS Transaction Server for OS/390®.

**IMS™** Represents the IMS Database Manager or IMS Transaction Manager.

**MVS™** Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

**RACF®**  
Represents the functions that are provided by the RACF component of the z/OS Security Server.

---

## Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in z/OS products, including DB2 UDB for z/OS, enable users to:

- Use assistive technologies such as screen reader and screen magnifier software
- Operate specific or equivalent features by using only a keyboard
- Customize display attributes such as color, contrast, and font size

Assistive technology products, such as screen readers, function with the DB2 UDB for z/OS user interfaces. Consult the documentation for the assistive technology products for specific information when you use assistive technology to access these interfaces.

Online documentation for Version 8 of DB2 UDB for z/OS is available in the DB2 Information Center, which is an accessible format when used with assistive technologies such as screen reader or screen magnifier software. The DB2 Information Center for z/OS solutions is available at the following Web site:  
<http://publib.boulder.ibm.com/infocenter/db2zhelp>.

---

## How to send your comments

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

- Send your comments by e-mail to [db2pubs@vnet.ibm.com](mailto:db2pubs@vnet.ibm.com) and include the name of the product, the version number of the product, and 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).
- You can also send comments from the Web. Visit the library Web site at:

[www.ibm.com/software/db2zos/library.html](http://www.ibm.com/software/db2zos/library.html)

This Web site has a feedback page that you can use to send comments.

- Print and fill out the reader comment form located at the back of this book. You can give the completed form to your local IBM branch office or IBM representative, or you can send it to the address printed on the reader comment form.



---

## Summary of changes to this book

Changes to this version of the book include:

- DDM command enhancements that reflect new functionality that is specified in The Open Group's DRDA V3 specification
- New DDM commands: EXCSQLGET, INTRDBRQS, and SNDPKT
- Restructuring of the DDM commands section in chapter 1 for easier readability

Technical updates to commands are marked with revision bars.



---

## Chapter 1. DDM command support

A requester using DRDA to connect to an application server or database server uses *Distributed Data Management (DDM)* as part of the underlying architecture of DRDA. DDM is the data connectivity language that is used for data interchange among like or unlike systems. DDM is independent of a particular system's hardware architecture and its operating system.

This chapter describes the DDM commands, command parameters, command and reply data objects, and reply messages that DB2 supports for DRDA levels 1, 2, 3, 4, and a subset of 5.

The IBM implementation of DRDA provides the following support for DDM:

- DRDA level 1 supports DDM level 3. This includes support for:
  - Base SQL
  - Application-directed connections
  - Remote unit of work connections
- DRDA level 2 supports DDM level 4. This includes support for:
  - Distributed unit of work connections
  - Protected connections (using SNA two-phase commit protocols)
- DRDA level 3 supports DDM level 5. This includes support for:
  - Objects and distinct types
  - Stored procedures with multi-row result sets
  - TCP/IP connections
  - Protected connections (using DRDA two-phase commit protocols)
  - Enhanced security (DRDA authentication security mechanisms)
  - Generic bind options
- DRDA level 4 supports DDM levels 5 and 6.
  - DDM level 5 includes support for:
    - Scrollable cursors
    - Connection and transaction pooling
    - Global transactions
    - Monitoring
    - Security enhancements (Kerberos authentication)
  - DDM level 6 includes support for:
    - Describe input
    - Database-directed access
    - Security enhancements (encrypted passwords)
    - New SQL data types (8-byte integer and datalink)
- DB2 also supports a subset of DRDA level 5, which provides support for the following DDM level 7 function.
  - Long SQL statements
  - SQL long identifiers
  - Query instance identifiers
  - Cursor extensions
  - Cursor attributes on prepare
  - Scrollable cursors
  - Multi-row fetch
  - Multi-dimensional multi-row insert
  - LOB processing enhancements
  - Keep dynamic package bind options
  - Data stream encryption
  - Extended describe
  - Interrupt query request
  - System monitoring

- Package path special register
- Extended diagnostics

This support is based on the September, 2003 draft of the Open Group Technical Standard DRDA Version 3. See The Open Group's Database Interoperability (DBIOP) Consortium Web page at <http://www.opengroup.org/dbiop/> for complete details and a current working draft of the specification.

---

## DRDA 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. SNA LU 6.2 also specifies the token (LUWID) that correlates the application process at the requester with the agent process at the server. 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 that are sent from a requester to a database server and all reply messages that are sent from a database server to a requester.

The DDM exchange server attributes (EXCSAT) command negotiates the level of DRDA that is supported. The DDM manager level information that is exchanged identifies the level of DRDA function that the requester and database server support. Table 1 on page 3 shows the DDM manager levels.
- Formatted Data Object Content Architecture (FD:OCA) describes the syntax and semantics of all command data objects that are sent from a requester to an application server, and all reply data objects that are sent from an application server to a requester.

Command data objects and reply data objects are defined by DDM; however, their structure is defined by FD:OCA.
- Character Data Representation Architecture (CDRA) specifies the representation of character data that is sent within a command data object from a requester to an application server, and character data that is sent within a reply data object from an application server to a requester.

Figure 1 on page 3 illustrates the relationship between DDM and other architectures that comprise DRDA.

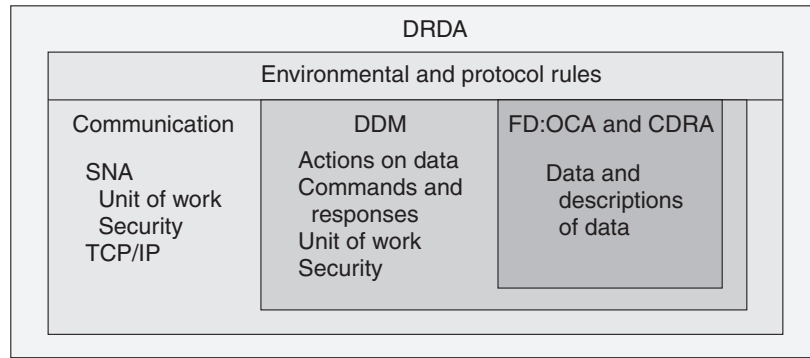


Figure 1. The relationship between DDM and the other architectures that comprise DRDA

The following table lists the DDM managers and the levels of function supported by DB2.

Table 1. DDM manager levels

Manager	Description	Minimum Level	Maximum Level
AGENT	Routes requests and replies	3	7 <sup>1</sup>
CMNAPPC	SNA communications manager	3	3
CMNSYNCPT	SNA sync point services	4	4
CMNTCPIP	TCP/IP communications manager	5	5
RDB	Relational database manager	3	3
RSYNMGR	Resynchronization manager	5	5
SECMGR	Security manager	5	7
SQLAM	SQL application manager	3	6 <sup>2</sup>
SYNCPTMGR	Sync point manager	5	5
XAMGR	XA manager	7	7

**Notes:**

1. Implies support for DRDA level 7 transaction pooling.
2. DRDA level 7 scrollable cursor requires special agreements between DB2 and DRDA requester vendors. Contact an IBM representative to request that DB2 enable support for your chosen DRDA vendor.

---

## Data type support

This section describes the support for DRDA data types.

### Large object SQL data types

DRDA supports the following large object (LOB) SQL data types. These data types are fully supported and are accessible by any DB2 Version 8 application or by a DB2 Version 8 server when connected to a DRDA level 4-compliant server or requester.

Large object bytes	Nullable large object bytes
Large object character SBCS	Nullable large object character SBCS
Large object character DBCS	Nullable large object character DBCS
Large object character mixed	Nullable large object character mixed
Large object bytes locator	Nullable large object bytes locator
Large object character locator	Nullable large object character locator
Large object character DBCS locator	Nullable large object character DBCS locator

## Row identifier data types

DRDA supports the following data types for row identifiers.

- Row identifier
- Nullable row identifier

## Eight-byte integer data types

DRDA supports the following data types for 8-byte integers. These data types are fully supported using compatible data types.

- 8-byte integer
- Nullable 8-byte integer

If a DB2 application accesses an 8-byte integer on a DRDA server that fully supports 8-byte integers, the DB2 requester converts the description and value of the 8-byte integer to a DECIMAL(19,0). If a DRDA requester provides an 8-byte integer in an input host variable, the DB2 server converts it to a DECIMAL(19,0) before processing the SQL statement.

## Datalink data types

DRDA supports the following data types for datalinks. These data types are fully supported using compatible data types.

- SBCS datalink
- Nullable SBCS datalink
- Mixed-byte (MBCS) datalink
- Nullable mixed-byte (MBCS) datalink

If a DB2 application accesses a datalink on a DRDA server that fully supports datalinks, the DB2 requester converts the description and value to a LONG VARCHAR. The application is required to parse the contents of the datalink data type. See *Open Group Technical Standard, DRDA Version 3 Vol. 1: Distributed Relational Database Architecture* for a complete description of the contents of a datalink data type. If a DRDA requester provides a datalink in an input host variable, the DB2 server converts it to a LONG VARCHAR before processing the SQL statement.

---

## Bookmarks with scrollable cursor support

For interfaces that require bookmark support, DB2 UDB for z/OS returns a count of the rows in the result table for a scrollable cursor. Table 2 on page 5 shows when the row count is returned to the application.

Table 2. When row count is returned

<b>FETCH specification</b>	<b>Resulting cursor position: On last row</b>	<b>Resulting cursor position: After last row</b>
FETCH NEXT	if on 'next to last row' SQLCODE 0 else n/a	if already on 'last row' SQLCODE +100, rowcount=n
FETCH PRIOR	if already 'after the last row' SQLCODE 0 else n/a	n/a
FETCH FIRST	if only one row SQLCODE 0 else n/a	if no rows SQLCODE +100, rowcount=0 else n/a
FETCH LAST	if one or more rows SQLCODE 0, rowcount=n else n/a	if no rows SQLCODE +100, rowcount=0 else n/a
FETCH BEFORE	n/a	n/a
FETCH AFTER	n/a	SQLCODE 0 <sup>1</sup> rowcount=n
FETCH CURRENT	if already on 'last' row SQLCODE 0 else n/a	if already 'after the last row' SQLCODE +231 else n/a
FETCH ABSOLUTE 1	if only one row SQLCODE 0 else n/a	if no rows SQLCODE +100, rowcount=0 else n/a
FETCH ABSOLUTE -1	if one or more rows SQLCODE 0, rowcount=n else n/a	n/a (if no rows, cursor position would be 'before')
FETCH ABSOLUTE n <sup>2</sup>	SQLCODE 0	n/a
FETCH ABSOLUTE -n <sup>2</sup>	if only one row SQLCODE 0 else n/a	n/a
FETCH ABSOLUTE x <sup>3</sup>	n/a	SQLCODE +100 rowcount=n
FETCH ABSOLUTE -x <sup>3</sup>	n/a	n/a (cursor position would be 'before')
FETCH RELATIVE r <sup>2</sup>	if currently r rows from the last one SQLCODE 0	if r is more than the remaining rows to the last one SQLCODE +100, rowcount=n
FETCH RELATIVE -r <sup>2</sup>	if r is 1 (FETCH RELATIVE -1) SQLCODE 0 else n/a	n/a (if no rows, cursor position would be 'before')

**Notes:**

1. On FETCH AFTER, the SQLCODE is zero for compatibility with DB2 UDB for iSeries. This is different than FETCH ABSOLUTE x which returns SQLCODE +100.
2. n is the number of rows of the result table and n>1 (where n=1 is handled by separate rows in the table for ABSOLUTE 1 and ABSOLUTE -1).
3. x > the number of rows of the result table and x>1 (where x=1 is handled by separate rows in the table for ABSOLUTE 1 and ABSOLUTE -1).

---

## Query processing rules

A query block is the basic unit of transmission for query and result set data. A requester can specify the size of query blocks in the *qryblksz* instance variable of the commands that can return data (OPNQRY, EXCSQLSTT, and CNTQRY). Specifying the query block size enables the requester, which may have resource constraints, to control the amount of data that is returned at any one time.

DRDA defines two types of blocking:

- Exact blocking—every query block must be exactly the same size. The only exception to this rule is the last query block in the reply chain, which can be smaller.
- Flexible blocking—each query block can be a different size, depending on the size of the row or result set being returned. The specified query block size is used as an initial size, and the query block can expand beyond that size, if necessary to complete the fetch operation.

As a requester, DB2 supports both exact and flexible blocking; as a server, DB2 supports (sends) only flexible blocking.

---

## Monitoring

Monitoring returns the database server's elapsed time as reply data that is chained to the normal reply. The following SQL-related commands support monitoring:

- CLSQRY
- CNTQRY
- DSCRDBTBL
- DSCSQLSTT
- EXCSQLIMM
- EXCSQLSET
- EXCSQLSTT
- OPNQRY
- PRPSQLSTT

---

## How to read the DDM command tables

DB2 supports all of the DDM commands described in this book.

A Distributed Data Management (DDM) command can have associated with it:

- Command parameters (instance variables)
- Command data objects
- Reply data objects
- Reply messages, containing reply message instance variables

DDM commands, and the objects and messages associated with them, map to actions taken by a relational database management system:

- Requesters send SQL requests to the database server as DDM commands.
- Requesters send host variable descriptions and values to the database server as command data objects.
- Database servers return data and a description of that data to requesters in the form of reply data objects.
- Database servers return information about the outcome of SQL requests to requesters in the form of reply messages.



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.

The **Required?** column in all of the tables indicates whether the item is required by DRDA. If the item is required, the column contains a value of Y; if the item is optional, the column contains a value of N. The meanings of the **Requester** and **Server** columns in each table are described in the sections below.

## Command parameters

This section describes the meaning of the **Requester** and **Server** columns in each command instance variable table.

The **Requester** column indicates whether DB2, as a requester, supports the command instance variable.

**Y** Means that DB2 supports the instance variable. DB2 provides a value for the instance variable each time it issues the command, assuming that a value exists. (Values might not exist for optional variables.)

**N** Means that DB2 does not support the instance variable. DB2 never provides this information when it issues the command.

**S** Means that DB2 supports a subset of the instance variable's options. The options are listed directly below the instance variable in capital letters. For each option, Y means that DB2 supports it, and N means that DB2 does not support it.

**N/A** Means that support for the instance variable is not applicable because DB2 never sends the command.

The **Server** column indicates whether DB2, as a server, supports the command instance variable.

**Y** Means that DB2 supports the instance variable. DB2 recognizes and processes this information, if received.

**N** Means that DB2 does not support the instance variable. DB2 rejects this information, if received.

**I** Means that DB2 does not support the instance variable. DB2 ignores this information, if received.

**S** Means that DB2 supports a subset of the instance variable's options. The options are listed directly below the instance variable in capital letters. For each option, Y means that DB2 supports it, and N means that DB2 rejects it.

## Command data objects

This section describes the meaning of the **Requester** and **Server** columns in each command data object table.

The **Requester** column indicates whether DB2, as a requester, supports the data object.

**Y** Means that DB2 supports the data object. DB2 provides a value for the object each time it issues the command, assuming that a value exists. (Values might not exist for optional objects.)

**N** Means that DB2 does not support the data object. DB2 never provides a value for the object when it issues the command.

I            **N/A**    Means that support for the data object is not applicable because DB2 never  
I            sends the command.

The **Server** column indicates whether DB2, as a server, supports the data object.

**Y**            Means that DB2 supports the data object. DB2 recognizes and processes  
this information, if received.

**I**            Means that DB2 does not support the data object. DB2 ignores this  
information, if received.

## Reply data objects

This section describes the meaning of the **Requester** and **Server** columns in each  
reply data object table.

I            The **Requester** column indicates whether DB2, as a requester, supports the data  
I            object.

**Y**            Means that DB2 supports the data object. DB2 recognizes and processes  
the information, if received.

**I**            Means that DB2 does not support the data object. DB2 ignores this  
information, if received.

I            **N**            Means that DB2 does not support the data object. DB2 rejects this  
I            information, if received.

I            **N/A**        Means that support for the data object is not applicable because DB2 never  
I            sends the command and thus, never receives the data object in response.

I            The **Server** column indicates whether DB2, as a server, supports the data object.

**Y**            Means that DB2 supports the data object. DB2 provides a value for the  
object each time it sends a reply, assuming that a value exists. (Values  
might not exist for optional objects.).

**N**            Means that DB2 does not support the data object. DB2 never provides a  
value for the object when it sends a reply.

## Reply messages

DDM reply messages fall into two classes:

- Normal response

This class represents the reply messages that are returned in response to the  
normal processing of a DDM command.

- Error response

This class represents the reply messages that are returned in response to an  
error detected by the application server during the processing of a DDM  
command.

### Normal situation

This section describes the meaning of the **Requester** and **Server** columns in each  
reply message instance variable table.

I            The **Requester** column indicates whether DB2, as a requester, supports the reply  
I            message instance variable.

**Y**            Means that DB2 supports the instance variable. DB2 recognizes and  
processes the information, if received.

**N** Means that DB2 does not support the instance variable. DB2 rejects the information, if received.

The **Server** column indicates whether DB2, as a server, supports the reply message instance variable.

**Y** Means that DB2 supports the instance variable. DB2 provides a value for the variable each time it sends a reply, assuming that a value exists.

**N** Means that DB2 does not support the instance variable. DB2 never provides a value for the variable when it sends a reply.

### **Error situation**

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

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

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

CMDATHRM	RDBATHRM	TRGNSPRM
----------	----------	----------

---

## ACCRDB command

The ACCRDB (access relational database) command establishes a path to the specified relational database (RDB).

### Command parameters

The ACCRDB command has the following instance variables.

Table 3. ACCRDB command instance variables

ACCRDB command instance variable	Requester	Server	Required?
rdbnam (RDB name)	Y	Y	Y
rdbaccl (RDB 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 allows updates)	Y	Y	N
prddta (product-specific data) <sup>1</sup>	Y	Y	N
ststrdel (statement string delimiter)	S	Y	N
STRDELAP	Y	Y	
STRDELQ	Y	Y	
DFTPKG	N	Y	
stdecdel (statement decimal delimiter)	S	Y	N
DECDELPRD	Y	Y	
DECDELCMA	Y	Y	
DFTPKG	N	Y	
crrtkn (correlation token)	Y	Y	Y
trgdfrt (target default values return)	N	Y	N
diaglvl (diagnostic level)	Y	Y	N
armcorr (ARM correlator)	N	Y	N

**Note:** <sup>1</sup> See Chapter 2, "Accounting for distributed data," on page 53 for more information.

### Command data objects

DRDA defines no command data objects for the ACCRDB command.

### Reply data objects

The ACCRDB command has the following reply data object.

Table 4. ACCRDB reply data object

ACCRDB reply data object	Requester	Server	Required?
sqlcard (SQLCA reply data)	Y	N/A	N

### Reply messages

The ACCRDBRM (access to RDB completed) reply message has the following instance variables.

Table 5. Instance variables of the ACCRDBRM reply message

ACCRDBRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
prdid (product-specific identifier)	Y	Y	Y
typdefnam (data type definition name)	Y	Y	Y
typdefovr (type definition overrides)	Y	Y	Y
rdbinttkn (RDB interrupt token)	Y	N	N
crrtkn (correlation token)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
pkgdfcst (package default character subtype)	Y	N	N
usrld (user ID at the target system)	Y	N	N
svlst (server list) <sup>1</sup>	Y	Y	N
ipaddr (TCP/IP address)	Y	Y	N
snaaddr (SNA address)	Y	N	N

**Note:** <sup>1</sup> svlst is supported by DRDA level 3 or higher.

---

## ACCSEC command

The ACCSEC (access security) command initializes the security mechanism that is used to validate the user's identity.

### Command parameters

The ACCSEC command has the following instance variables.

Table 6. ACCSEC command instance variables

ACCSEC command instance variable	Requester	Server	Required?
secmgrnm (security manager name)	N	I	N
secmec (security mechanism) <sup>1</sup>	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
sectkn (security token)	N	Y	N
encalg (encryption algorithm)	N	Y	N
enckeylen (encryption key length)	N	Y	N

**Note:** <sup>1</sup> See Table 7.

### Security mechanisms

Table 7 lists the security mechanisms that DB2 supports, and indicates which of the security mechanisms are used by DB2 requesters and which of the security mechanisms are accepted by DB2 servers.

Table 7. Security mechanisms supported by DB2

SECMEC	Requesters use	Servers accept	Description
X'0003'	Y	Y	User ID password
X'0004'	Y	Y	User ID only
X'0005'	N	Y	Change password
X'0006'	N	N	User ID, password substitute
X'0007'	N	Y	User ID, encrypted password
X'0009'	Y	Y	Encrypted user ID and password
X'000A'	N	Y	Encrypted change password
X'000B'	N	Y	Kerberos
X'000C'	Y	Y	Encrypted user ID and data
X'000D'	Y	Y	Encrypted user ID, password, and data
X'000E'	N	Y	Encrypted user ID, password, new password, and data

### Command data objects

DRDA defines no command data objects for the ACCSEC command.

### Reply data objects

The ACCSECRD (access security) reply data object has the following instance variables.

Table 8. Instance variables of the ACCSECRD reply data object

<b>ACCSECRD reply data object instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
sectkn (security token)	N	Y	N
secmec (security mechanism)	Y	Y	Y
secchkcd (security check code)	N	Y	N
encalg (encryption algorithm)	N	Y	N
enckeylen (encryption key length)	N	Y	N

## Reply messages

Normal processing of the ACCSEC command results in no DDM reply messages.

---

## BGNBND command

The BGNBND (begin bind) command starts the process of binding a package to the specified relational database (RDB).

### Command parameters

The BGNBND command has the following instance variables.

Table 9. BGNBND command instance variables

BGNBND command instance variable	Requester	Server	Required?
rdbnam (RDB name, 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 <sup>1</sup>	Y	Y	
DEFINER_REVERT_TO_OWNER <sup>1</sup>	Y	Y	
INVOKER_REVERT_TO_REQUESTER <sup>1</sup>	Y	Y	
INVOKER_REVERT_TO_OWNER <sup>1</sup>	Y	Y	
ststrdel (statement string delimiter)	Y	Y	N
sttdecdel (statement decimal delimiter)	Y	Y	N
sttdatfmt (statement date format)	Y	Y	N
stttimefmt (statement time format)	Y	Y	N
pkgisolvl (package isolation level)	Y	Y	Y
ISOLVLR	Y	Y	
ISOLVLALL	Y	Y	
ISOLVLCS	Y	Y	
ISOLVLCHG	Y	Y	
ISOLVLNC	Y	Y	
dgriopr (degree of I/O parallelism)	Y	Y	N
bndcrtctl (bind package 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
dftrdbc (default RDB collection identifier)	Y	Y	N
title (brief description of package)	N	I	N
qryblkctl (query block protocol control)	Y	Y	N



Table 9. BGNBND command instance variables (continued)

BGNBND command instance variable	Requester	Server	Required?
pkgdfcst (package default character subtype)	N	S	N
CSTSYDFT	N	Y	
CSCTBITS	N	I	
CSTSBCS	N	I	
CSTMBCS	N	I	
pkgdfcc (package default CCSIDs)	N	Y	N
pkgrpivr (replaced package version name)	Y	Y	Y
decprc (decimal precision)	N	Y	N
prpstkp (prepared statement keep)	N	Y	N

**Note:** <sup>1</sup> These options are supported only by DRDA level 4 and above.

## Command data objects

The BGNBND command has the following command data object. (The list under the command data object shows the generic bind options that DB2 supports (sends and receives). Allowed values are shown in parentheses, if applicable.)

Table 10. BGNBND command data object

BGNBND command data object	Requester	Server	Required?
bndopt (bind option) <sup>1</sup>	S	Y	N
DBPROTOCOL(DRDA   PRIVATE) <sup>2</sup>	Y	Y	
DEFER(PREPARE) <sup>2</sup>	Y	Y	
NODEFER(PREPARE) <sup>2</sup>	Y	Y	
ENCODING(ASCII   EBCDIC   UNICODE   <i>ccsid</i> ) <sup>2</sup>	Y	Y	
FUNCPATH( <i>schema-name</i> )	N	Y	
IMMEDWRITE(NO   YES   PH1) <sup>2</sup>	Y	Y	
KEEPDYNAMIC(NO   YES) <sup>2</sup>	N	Y	
LANGUAGE(COBOL   COBOL2   IBMCOBOL) <sup>3</sup>	Y	Y	
MINDIVSCALE( <i>number-char</i> )	Y	Y	
OPTHINT( <i>hint-id</i> ) <sup>2</sup>	Y	Y	
PATH( <i>schema-name</i> ) <sup>2</sup>	Y	Y	
PATHDEFAULT <sup>2</sup>	Y	Y	
PRECOMPCCSID( <i>number-string</i> ) <sup>3</sup>	Y	Y	
REOPT(VARS) <sup>2</sup>	Y	Y	
NOREOPT(VARS) <sup>2</sup>	Y	Y	

**Notes:**

1. bndopt is supported by DRDA level 3 only.
2. See *DB2 Command Reference* for complete information about these bind options.
3. These options flow only between z/OS requesters and z/OS servers.

## Reply data objects

The BGNBND command has the following reply data objects.

*Table 11. BGNBND reply data objects*

<b>BGNBND reply data object</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
mgrlvlvovr (manager level overrides)	Y	N	N

## Reply messages

The RDBUPDRM (RDB update) reply message has the following instance variables. (The RDBUPDRM reply message was introduced in DRDA level 2.)

*Table 12. Instance variables of the RDBUPDRM reply message*

<b>RDBUPDRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

---

## BNDSQLSTT command

The BNDSQLSTT (bind SQL statement) command binds an SQL statement to a relational database (RDB) package.

### Command parameters

The BNDSQLSTT command has the following instance variables.

Table 13. BNDSQLSTT command instance variables

BNDSQLSTT command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
sqlstnbr (SQL statement number)	Y	Y	N
bdnstasm (bind SQL statement assumptions)	Y	Y	N
pkgsn (RDB package section number)	N	N	N

### Command data objects

The BNDSQLSTT command has the following command data objects.

Table 14. BNDSQLSTT command data objects

BNDSQLSTT command data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlstt (SQL statement to bind)	Y	Y	Y
sqlsttvr (SQL statement variable descriptions)	Y	Y	N

**Note:** <sup>1</sup> sectknovr must precede sqlstt. sectknovr is supported only when intermediate server processing (hopping) is involved.

### Reply data objects

The BNDSQLSTT command has the following reply data objects.

Table 15. BNDSQLSTT reply data objects

BNDSQLSTT reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
mgrlvovr (manager level overrides)	Y	N	N

### Reply messages

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

*Table 16. Instance variables of the RDBUPDRM reply message*

<b>RDBUPDRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

I

---

## CLSQRV command

The CLSQRV (close query) command closes a query that was previously opened by an OPNQRV command.

### Command parameters

The CLSQRV command has the following instance variables.

Table 17. CLSQRV command instance variables

CLSQRV command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
monitor (monitor events)	N	Y	N
cmdsrcid (command source identifier)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	Y
qryclsrls (query close lock release)	N	Y	N
pkgsn (RDB package section number)	N	N	N

### Command data objects

DRDA defines no command data objects for the CLSQRV command.

### Reply data objects

The CLSQRV command has the following reply data objects.

Table 18. CLSQRV reply data objects

CLSQRV reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

### Reply messages

Normal processing of the CLSQRV command results in no DDM reply messages.

---

## CNTQRY command

The CNTQRY (continue query) command resumes a query or resumes the return of result set data that was interrupted.

### Command parameters

The CNTQRY command has the following instance variables.

Table 19. CNTQRY command instance variables

CNTQRY command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (RDB package name, consistency token, and section number)	Y	Y	N
qryblksz (query block size)	Y	Y	Y
maxblkext (maximum number of additional query blocks) <sup>1</sup>	Y	Y	N
monitor (monitor events)	N	Y	N
rtnextdta (return of extdta option) <sup>1</sup>	Y	Y	N
RTNEXTROW (return EXTDTAs by row)	Y	Y	
RTNEXTALL (return all EXTDTAs for qrydta sent) <sup>2</sup>	Y	Y	
qryblkrst (query block reset)	Y	Y	N
qryrownbr (query row number)	Y	Y	N <sup>3</sup>
qryrowset (query row set size)	Y	Y	N <sup>3</sup>
qryrowsns (query row sensitivity)	Y	Y	N
qyrtdta (query returns data)	Y	Y	N
qyscrorn (query scroll orientation)	Y	Y	N
cmdsrcid (command source identifier)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	Y
pkgsn (RDB package section number)	N	N	N

**Notes:**

1. maxblkext and rtnextdta are supported only by DRDA level 4 and above.
  2. The requester supports RTNEXTALL for scrollable cursors only.
  3. qryrownbr and qryrowset are required under some circumstances. Refer to The Open Group DRDA specification for complete details.
- 

### Command data objects

The CNTQRY command has the following command data object.

Table 20. CNTQRY command data object

CNTQRY command data object	Requester	Server	Required?
outovr (output override descriptor)	Y	Y	N

## Reply data objects

The CNTQRY command has the following reply data objects.

Table 21. CNTQRY reply data objects

CNTQRY reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
qrydta (query answer set data)	Y	Y	N
extdta (externalized FD:OCA data) <sup>2</sup>	Y	Y	N
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

### Notes:

1. sectknovr must precede qrydta. sectknovr is supported only when intermediate server processing (hopping) is involved.
2. extdta is supported only by DRDA level 4 and above.

## Reply messages

Any of the following reply messages can be returned from a CNTQRY command.

### ENDQRYRM

The ENDQRYRM (end of query condition) reply message has the following instance variables.

Table 22. Instance variables of the ENDQRYRM reply message

ENDQRYRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svrdgn (server diagnostic information)	Y	Y	Y

### RDBUPDRM

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 23. Instance variables of the RDBUPDRM reply message

RDBUPDRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y

---

## DRPPKG command

The DRPPKG (drop package) command drops the specified package from a relational database (RDB).

### Command parameters

The DRPPKG command has the following instance variables.

Table 24. DRPPKG command instance variables

DRPPKG command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnam (package grouping name and identifier)	Y	Y	Y
vrsnam (version name)	Y	Y	N

### Command data objects

DRDA defines no command data objects for the DRPPKG command.

### Reply data objects

The DRPPKG command has the following reply data objects.

Table 25. DRPPKG reply data objects

DRPPKG reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
mgrlvlvr (manager level overrides)	Y	N	N

### Reply messages

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 26. RDBUPDRM reply message instance variables

RDBUPDRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y



---

## DSCRDBTBL command

The DSCRDBTBL (describe relational database table) command requests that a description of the specified table (named in the *sqltblnam* command data object) be returned.

DB2 only sends the DSCRDBTBL command when it is acting as a server and then, only when it is propagating an interrupt RDB request to a downstream hop server. (A hop server is a server that acts as a requester to another server.)

## Command parameters

The DSCRDBTBL command has the following instance variables.

Table 27. DSCRDBTBL command instance variables

DSCRDBTBL command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	Y	Y	N
monitor (monitor events)	N	Y	N

## Command data objects

The DSCRDBTBL command has the following command data objects.

Table 28. DSCRDBTBL command data objects

DSCRDBTBL command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	Y	Y	N
sqltblnam (SQL table name)	Y	Y	Y

## Reply data objects

The DSCRDBTBL command has the following reply data objects.

Table 29. DSCRDBTBL reply data objects

DSCRDBTBL reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
sqldard (SQLDA reply data)	Y	Y	Y
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	N	Y

**Note:** <sup>1</sup> sectknovr must precede sqldard. sectknovr is supported only when intermediate server processing (hopping) is involved.

## Reply messages

Normal processing of the DSCRDBTBL command results in no DDM reply messages.

---

## DSCSQLSTT command

The DSCSQLSTT (describe SQL statement) command returns the column definitions of the result table of a prepared or bound statement in addition to the names and labels of those columns.

DB2 sends a DSCSQLSTT command only when executing the SQL DESCRIBE INPUT statement.

## Command parameters

The DSCSQLSTT command has the following instance variables.

Table 30. DSCSQLSTT command instance variables

DSCSQLSTT command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
typsqlda (type of descriptor to return)	S	Y	N
0 (return output SQLDA, default)	N	Y	
1 (return input SQLDA)	Y	Y	
monitor (monitor events)	N	Y	N
cmdsrid (command source identifier)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	N
pkgcn (RDB package section number)	N	N	N

## Command data objects

DRDA defines no command data objects for the DSCSQLSTT command.

## Reply data objects

The DSCSQLSTT command has the following reply data objects.

Table 31. DSCSQLSTT reply data objects

DSCSQLSTT reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
sqldard (SQLDA reply data)	Y	Y	Y
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

**Note:** <sup>1</sup> sectknovr must precede sqldard. sectknovr is supported only when intermediate server processing (hopping) is involved.

## Reply messages

Normal processing of the DSCSQLSTT command results in no DDM reply messages.

---

## ENDBND command

The ENDBND (end bind) command terminates the process of binding a package to a relational database (RDB). This command indicates that no more BIND commands are to be sent; the package is now complete.

### Command parameters

The ENDBND command has the following instance variables.

Table 32. ENDBND command instance variables

ENDBND command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamct (package name and consistency token)	Y	Y	Y
maxsctnbr (maximum section number)	Y	Y	N

### Command data objects

DRDA defines no command data objects for the ENDBND command.

### Reply data objects

The ENDBND command has the following reply data objects.

Table 33. ENDBND reply data objects

ENDBND reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
mgrlvlivr (manager level overrides)	Y	N	N

### Reply messages

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 34. Instance variables of the RDBUPDRM reply message

RDBUPDRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y

---

## EXCSAT command

The EXCSAT (exchange server attributes) command is the first DDM command that is sent from a requester to a server. Using this command, the requester communicates the minimum level of support that it requires from the server.

### Command parameters

The EXCSAT command has the following instance variables.

*Table 35. EXCSAT command instance variables*

<b>EXCSAT command instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
extnam (external name)	Y	Y	N
mgrlvlsls (manager level list)	Y	Y	N
spvnam (supervisor name)	N	N	N
svclsnm (server class name)	Y	Y	N
svnam (server name)	Y	Y	N
svrlslv (server release level)	Y	Y	N

### Command data objects

DRDA defines no command data objects for the EXCSAT command.

### Reply data objects

The EXCSATRD reply data object has the following instance variables.

*Table 36. Instance variables of the EXCSATRD reply data object*

<b>EXCSATRD reply data object instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
extnam (external name)	Y	Y	N
mgrlvlsls (manager level list)	Y	Y	N
svclsnm (server class name)	Y	Y	N
svnam (server name)	Y	Y	N
svrlslv (server release level)	Y	Y	N

### Reply messages

Normal processing of the EXCSAT command results in no DDM reply messages.

---

## EXCSQLGET command

The EXCSQLGET (execute SQL get) command retrieves the current value of the special register that is identified by the SQL object name.

As a server, DB2 accepts this command from any requester. As a requester, DB2 only sends this command when the server is a z/OS Version 8 DB2 server.

## Command parameters

The EXCSQLGET command has the following instance variables.

Table 37. EXCSQLGET command instance variables

EXCSQLGET command instance variable	Requester	Server	Required?
rdbnam (RDB name)	Y	Y	Y
monitor (monitor events)	N	Y	N

## Command data objects

The EXCSQLGET command has the following command data objects.

Table 38. EXCSQLGET command data objects

EXCSQLGET command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	Y	Y	N
sqlobjnam (SQL object name)	Y	Y	Y

## Reply data objects

The EXCSQLGET command has the following reply data objects.

Table 39. EXCSQLGET reply data objects

EXCSQLGET reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
monitorrd (monitor reply data)	N	Y	N
sqldtard (SQL data reply data)	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N

**Note:** <sup>1</sup> sectknovr must precede sqldtard. sectknovr is supported only when intermediate server processing (hopping) is involved.

## Reply messages

Normal processing of the EXCSQLGET command results in no DDM reply messages.

---

## EXCSQLIMM command

The EXCSQLIMM (execute immediate SQL statement) command executes the single SQL statement that is sent as command data.

### Command parameters

The EXCSQLIMM command has the following instance variables.

Table 40. EXCSQLIMM command instance variables

EXCSQLIMM command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
monitor (monitor events)	N	Y	N
cmdsrcid (command source identifier)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	N
pkgsn (RDB package section number)	N	N	N

### Command data objects

The EXCSQLIMM command has the following command data objects.

Table 41. EXCSQLIMM command data objects

EXCSQLIMM command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	N	Y	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlstt (SQL statement)	Y	Y	Y

**Note:** <sup>1</sup> sectknovr must precede sqlstt. sectknovr is supported only when intermediate server processing (hopping) is involved.

### Reply data objects

The EXCSQLIMM command has the following reply data objects.

Table 42. EXCSQLIMM reply data objects

EXCSQLIMM reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

### Reply messages

Any of the following reply messages can be returned from an EXCSQLIMM command.

## CMMRQSRM

The CMMRQSRM (commitment request) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 43. Instance variables of the CMMRQSRM reply message

CMMRQSRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
cmmtyp (commitment request type)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

## ENDUOWRM

The ENDUOWRM (end unit of work condition) reply message has the following instance variables.

Table 44. Instance variables of the ENDUOWRM reply message

ENDUOWRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
uowdsp (unit of work disposition)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
rlsconv (release connection)	N	Y	N

## RDBUPDRM

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 45. Instance variables of the RDBUPDRM reply message

RDBUPDRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

---

## EXCSQLSET command

The EXCSQLSET (execute SQL set) command executes one or more SET statements to establish the application environment.

### Command parameters

The EXCSQLSET command has the following instance variables.

Table 46. EXCSQLSET command instance variables

EXCSQLSET command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	Y
pkgnamct (RDB package name and consistency token)	Y	Y	N
monitor (monitor events)	N	Y	N

### Command data objects

The EXCSQLSET command has the following command data objects.

Table 47. EXCSQLSET command data objects

EXCSQLSET command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	N	Y	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlstt (SQL statement) <sup>2</sup>	Y	Y	Y

**Notes:**

1. sectknovr must precede sqlstt. sectknovr is supported only when intermediate server processing (hopping) is involved.
2. See Chapter 2, "Accounting for distributed data," on page 53 for more information.

### Reply data objects

The EXCSQLSET command has the following reply data objects.

Table 48. EXCSQLSET reply data objects

EXCSQLSET reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
monitorrd (monitor reply data)	N	Y	N
mgrlvlvovr (manager level overrides)	Y	N	N

### Reply messages

Normal processing of the EXCSQLSET command results in no DDM reply messages.



## EXCSQLSTT command

The EXCSQLSTT (execute SQL statement) command executes a previously bound SQL statement.

### Command parameters

The EXCSQLSTT command has the following instance variables.

Table 49. EXCSQLSTT command instance variables

EXCSQLSTT command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
prcnam (stored procedure name)	Y	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
outexp (output expected)	Y	Y	N
maxrslcnt (maximum number of result sets) <sup>1</sup>	Y	Y	N
maxblkext (maximum number of additional query blocks) <sup>2</sup>	Y	Y	N
qryblksz (query block size)	N	Y	Y
rssetflg (result set flags)	Y	Y	N
rdbcmtok (RDB commit by application server allowed) <sup>3</sup>	N	Y	N
outovropt (output override option) <sup>2</sup>	Y	Y	N
OUTOVRFRS (override allowed on first CNTQRY)	Y	Y	
OUTOVRANY (override allowed on any CNTQRY)	Y	Y	
qyrowset (query row set size)	Y	Y	N
monitor (monitor events)	N	Y	N
cmdsrcid (command source identifier)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	N
pkgsn (RDB package section number)	N	N	N
typsqlda (type of SQL descriptor to return)	N	Y	N

#### Notes:

1. maxrslcnt is supported by DRDA level 3 and higher.
2. maxblkext and outovropt are supported by DRDA level 4 and higher.
3. rdbcmtok is supported by DRDA level 2 and higher.

### Command data objects

The EXCSQLSTT command has the following command data objects.

Table 50. EXCSQLSTT command data objects

EXCSQLSTT command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	N	Y	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqllda (SQL program variable data)	Y	Y	N

Table 50. EXCSQLSTT command data objects (continued)

EXCSQLSTT command data object	Requester	Server	Required?
prcnam (stored procedure name)	Y	Y	N
extdta (externalized FD:OCA data)	Y	Y	N
outovr (output override descriptor) <sup>2</sup>	Y	Y	N

**Notes:**

- sectknovr must precede the first encrypted command data object. sectknovr is supported only when intermediate server processing (hopping) is involved.
- outovr is supported only by DRDA level 4 and above.

## Reply data objects

The EXCSQLSTT command has the following reply data objects.

Table 51. EXCSQLSTT reply data objects

EXCSQLSTT reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
sqldtard (SQL data reply data)	Y	Y	N
sqlcinrd (SQL result set column information reply data)	Y	Y	N
sqlrslrd (SQL result set reply data) <sup>2</sup>	Y	Y	N
qrydsc (query answer set description) <sup>2</sup>	Y	Y	N
qrydta (query answer set data) <sup>2</sup>	Y	Y	N
extdta (externalized FD:OCA data) <sup>3</sup>	Y	Y	N
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

**Notes:**

- sectknovr must precede the first encrypted reply data object. sectknovr is supported only when intermediate server processing (hopping) is involved.
- sqlrslrd, qrydsc, and qrydta are supported only by DRDA level 3.
- extdta is supported only by DRDA level 4 or above.

## Reply messages

Any of the following reply messages can be returned from an EXCSQLSTT command.

### CMMRQSRM

The CMMRQSRM (commitment request) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 52. Instance variables of the CMMRQSRM reply message

CMMRQSRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y

Table 52. Instance variables of the CMMRQSRM reply message (continued)

CMMRQSRM reply message instance variable	Requester	Server	Required?
cmmtyp (commitment request type)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

## ENDQRYRM

The ENDQRYRM (end of query condition) reply message has the following instance variables.

Table 53. Instance variables of the ENDQRYRM reply message

ENDQRYRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y

## ENDUOWRM

The ENDUOWRM (end unit of work condition) reply message has the following instance variables.

Table 54. Instance variables of the ENDUOWRM reply message

ENDUOWRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
uowdsp (unit of work disposition)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
rlsconv (release connection)	N	Y	N

## OPNQRYSRM

The OPNQRYSRM (open query completed) reply message has the following instance variables.

Table 55. Instance variables of the OPNQRYSRM reply message

OPNQRYSRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
qryprctyp (query protocol type)	Y	Y	Y
sqlcsrhd (hold cursor position)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
qryattscr (query attribute for scrollability)	Y	Y	N
qryattsns (query attribute for sensitivity)	Y	Y	N
qryattupd (query attribute for updatability)	Y	Y	N
qryinsid (query instance identifier)	Y	Y	Y
qryblkfct (query blocking factor)	Y	Y	N
qryattset (query attribute for row set)	Y	Y	N

## RDBUPDRM

The RDBUPDRM (RDB update) reply message has the following instance variables. (The RDBUPDRM reply message was introduced in DRDA level 2.)

*Table 56. Instance variables of the RDBUPDRM reply message*

<b>RDBUPDRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
svrnod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y

## RSLSETRM

The RSLSETRM (RDB result set) reply message has the following instance variables. (The RSLSETRM reply message instance variables are specific to DRDA level 3 only.)

*Table 57. Instance variables of the RSLSETRM reply message*

<b>RSLSETRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
svrnod (severity code)	Y	Y	Y
pkgslst (package section list)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y

---

## INTRDBRQS command

The INTRDBRQS (interrupt relational database request) command allows a requester to end a DDM command that is currently executing.

The requester must establish a connection on which to send the INTRDBRQS command that is separate from the connection on which it sent the ACCRDB command. A requester cannot send an INTRDBRQS command on the same connection on which it sent the ACCRDB command. A requester can, however, send multiple INTRDBRQS commands on the same connection. The requester uses the values of the *rdbintrkn* instance variable that were received in the ACCRDBRM reply messages of previous ACCRDB commands that it issued.

DB2 only sends the INTRDBRQS command when it is acting as a server and then, only when it is propagating an interrupt RDB request to a downstream hop server. (A hop server is a server that acts as a requester to another server.)

## Command parameters

The INTRDBRQS command has the following instance variables.

*Table 58. INTRDBRQS command instance variables*

INTRDBRQS command instance variable	Requester	Server	Required?
rdbintrkn (RDB interrupt token)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N

## Command data objects

DRDA defines no command data objects for the INTRDBRQS command.

## Reply data objects

DRDA defines no reply data objects for the INTRDBRQS command.

## Reply messages

The CMDCMPRM (command processing completed) reply message has the following instance variables.

*Table 59. Instance variables of the CMDCMPRM reply message*

CMDCMPRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	N

---

## OPNQRY command

The OPNQRY (open query) command opens a query to the specified relational database (RDB). The command is sent as a result of an SQL OPEN statement.

See “PRPSQLSTT command” on page 39 for information about a resource limit condition that can exist when the OPNQRY command is chained to the PRPSQLSTT command.

## Command parameters

The OPNQRY command has the following instance variables.

Table 60. OPNQRY command instance variables

OPNQRY command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
qryblksz (query block size)	Y	Y	Y
qryblkctl (query block protocol control)	N	Y	N
maxblkext (maximum number of additional query blocks) <sup>1</sup>	Y	Y	N
outovropt (output override option) <sup>1</sup>	Y	Y	N
OUTOVRFRS (override allowed on first CNTQRY)	Y	Y	
OUTOVRANY (override allowed on any CNTQRY)	Y	Y	
qryrowset (query row set size)	Y	Y	N
monitor (monitor events)	N	Y	N
qryclsimp (query close implicit)	N	Y	N
qryclsrls (query close lock release)	N	Y	N
cmdsrcid (command source identifier)	Y	Y	N
dupqryok (duplicate query allowed)	Y	Y	N
pkgsn (RDB package section number)	N	N	N
typsqlda (type of SQL descriptor to return)	N	Y	N

**Note:** <sup>1</sup> maxblkext and outovropt are supported only by DRDA level 4 and above.

## Command data objects

The OPNQRY command has the following command data objects.

Table 61. OPNQRY command data objects

OPNQRY command data object	Requester	Server	Required?
typdefnam (data type definition name)	N	Y	N
typdefovr (data type definition override)	N	Y	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqllda (SQL program variable data)	Y	Y	N
extlda (externalized FD:OCA data) <sup>2</sup>	Y	Y	N

Table 61. OPNQRY command data objects (continued)

OPNQRY command data object	Requester	Server	Required?
<b>Notes:</b>			
1. sectknovr must precede sqltda. sectknovr is supported only when intermediate server processing (hopping) is involved.			
2. exttda is supported only by DRDA level 4 and above.			

## Reply data objects

The OPNQRY command has the following reply data objects.

Table 62. OPNQRY reply data objects

OPNQRY reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
qrydsc (query answer set description)	Y	Y	Y
qrydta (query answer set data)	Y	Y	N
monitorrd (monitor reply data)	N	Y	N
mgrlvlovr (manager level overrides)	Y	Y	N

**Note:** <sup>1</sup> sectknovr must precede qrydta. sectknovr is supported only when intermediate server processing (hopping) is involved.

## Reply message

Any of the following reply messages can be returned from an OPNQRY command.

### ENDQRYRM

The ENDQRYRM (end of query condition) reply message has the following instance variables.

Table 63. Instance variables of the ENDQRYRM reply message

ENDQRYRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svrdgn (server diagnostic information)	Y	Y	Y

### OPNQRYRM

The OPNQRYRM (open query completed) reply message has the following instance variables.

Table 64. Instance variables of the OPNQRYRM reply message

OPNQRYRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
qryprctyp (query protocol type)	Y	Y	Y
sqlcsrhd (hold cursor position)	Y	Y	N
svrdgn (server diagnostic information)	Y	Y	Y

Table 64. Instance variables of the OPNQRYSM reply message (continued)

<b>OPNQRYSM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
qyattscl (query attribute for scrollability)	Y	Y	N
qyattsns (query attribute for sensitivity)	Y	Y	N
qyattupd (query attribute for updatability)	Y	Y	N
qyinsid (query instance identifier)	Y	Y	Y
qryblkfct (query blocking factor)	Y	Y	N
qyattset (query row set attribute)	Y	Y	N

## **RDBUPDRM**

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 65. Instance variables of the RDBUPDRM reply message

<b>RDBUPDRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svdgn (server diagnostic information)	Y	Y	Y



---

## PRPSQLSTT command

The PRPSQLSTT (prepare SQL statement) command dynamically binds a single SQL statement to a section number in an existing package in a relational database (RDB).

### Command parameters

The PRPSQLSTT command has the following instance variables.

Table 66. PRPSQLSTT command instance variables

PRPSQLSTT command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	N	Y	N
pkgnamcsn (package name, consistency token, and section number)	Y	Y	N
rtnsqlda (return SQL descriptor area)	Y	Y	N
monitor (monitor events)	N	Y	N
cmdsrid (command source identifier)	Y	Y	N
pkgsn (RDB package section number)	N	N	N
typsqlda (type of descriptor to return)	N	Y	N

### Command data objects

The PRPSQLSTT command has the following command data objects.

Table 67. PRPSQLSTT command data objects

PRPSQLSTT command data object	Requester	Server	Required?
typedefnam (data type definition name)	N	Y	N
typedefovr (data type definition override)	N	Y	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlstt (SQL statement)	Y	Y	Y
sqlattr (SQL statement attributes)	Y	Y	N

**Note:** <sup>1</sup> sectknovr must precede sqlstt. sectknovr is supported only when intermediate server processing (hopping) is involved.

### Reply data objects

The PRPSQLSTT command has the following reply data objects.

Table 68. PRPSQLSTT reply data objects

PRPSQLSTT reply data object	Requester	Server	Required?
typedefnam (data type definition name)	Y	N	N
typedefovr (data type definition override)	Y	N	N
sectknovr (sectkn overrides) <sup>1</sup>	Y	Y	N
sqlcard (SQLCA reply data)	Y	Y	N
sqldard (SQLDA reply data)	Y	Y	N
monitorrd (monitor reply data)	N	Y	N
mgrlvlvovr (manager level overrides)	Y	Y	N

Table 68. PRPSQLSTT reply data objects (continued)

PRPSQLSTT reply data object	Requester	Server	Required?
<b>Note:</b> <sup>1</sup> sectknovr must precede sqldard. sectknovr is supported only when intermediate server processing (hopping) is involved.			

If the DB2 UDB for z/OS predictive governing function is active, a PRPSQLSTT command that is sent to a DB2 application server can receive an *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 OPNQRV command. If an OPNQRV command is chained to a PRPSQLSTT command that receives warning SQLSTATE 01616, the chained OPNQRV receives an *sqlcard* with error SQLSTATE 57057 (SQLCODE -30002). A requester that detects SQLSTATE 57057 can take one of the following actions:

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

For additional 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 requester (WN rules) in *Open Group Technical Standard, DRDA Version 3 Vol. 1: Distributed Relational Database Architecture*.

## Reply messages

Normal processing of the PRPSQLSTT command results in no DDM reply messages.

---

## RDBCMM command

The RDBCMM (relational database commit) command commits the current unit of work (transaction).

### Command parameters

The RDBCMM command has the following instance variables.

Table 69. RDBCMM command instance variable

RDBCMM command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	Y	Y	N
rlsconv (release connection)	N	Y	N

### Command data objects

DRDA defines no command data objects for the RDBCMM command.

### Reply data objects

The RDBCMM command has the following reply data objects.

Table 70. RDBCMM reply data objects

RDBCMM reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
sqlstt (SQL statements)	N	Y	N
mgrlvlovr (manager level overrides)	Y	N	N

### Reply messages

Either of the following reply messages can be returned from an RDBCMM command.

#### **CMDVLTRM**

The CMDVLTRM (command violation) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 71. Instance variables of the CMDVLTRM reply message

CMDVLTRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y

#### **ENDUOWRM**

The ENDUOWRM (end unit of work condition) reply message has the following instance variables.

Table 72. Instance variables of the ENDUOWRM reply message

ENDUOWRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y

Table 72. Instance variables of the ENDUOWRM reply message (continued)

<b>ENDUOWRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
uowdsp (unit of work disposition)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
rlsconv (release connection)	N	Y	N

I

---

## RDBRLLBCK command

The RDBRLLBCK (relational database roll back) command rolls back the current unit of work (transaction).

### Command parameters

The RDBRLLBCK command has the following instance variables.

Table 73. RDBRLLBCK command instance variables

RDBRLLBCK command instance variable	Requester	Server	Required?
rdbnam (RDB name, as in ACCRDB)	Y	Y	N
rlsconv (release connection)	N	Y	N

### Command data objects

DRDA defines no command data objects for the RDBRLLBCK command.

### Reply data objects

The RDBRLLBCK command has the following reply data objects.

Table 74. RDBRLLBCK reply data objects

RDBRLLBCK reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
sqlstt (SQL statements)	N	Y	N
mgrlvlovr (manager level overrides)	Y	N	N

### Reply messages

Either of the following reply messages can be returned from an RDBRLLBCK command.

#### CMDVLTRM

The CMDVLTRM (command violation) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 75. Instance variables of the CMDVLTRM reply message

CMDVLTRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y

#### ENDUOWRM

The ENDUOWRM (end unit of work condition) reply message has the following instance variables.

Table 76. Instance variables of the ENDUOWRM reply message

ENDUOWRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y

Table 76. Instance variables of the ENDUOWRM reply message (continued)

<b>ENDUOWRM reply message instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
uowdsp (unit of work disposition)	Y	Y	Y
rdbnam (RDB name)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y
rlsconv (release connection)	N	Y	N

I

---

## REBIND command

The REBIND (rebind) command rebinds an existing package at the server. No new SQL statements are sent with REBIND; the SQL statements must have been bound previously at the server.

### Command parameters

The REBIND command has the following instance variables.

Table 77. REBIND command instance variables

REBIND command instance variable	Requester	Server	Required?
rdbnam (RDB name, 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 <sup>1</sup>	Y	Y	
DEFINER_REVERT_TO_OWNER <sup>1</sup>	Y	Y	
INVOKER_REVERT_TO_REQUESTER <sup>1</sup>	Y	Y	
INVOKER_REVERT_TO_OWNER <sup>1</sup>	Y	Y	
pkgisolvl (package isolation level)	Y	Y	Y
ISOLVLR	Y	Y	
ISOLVLR	Y	Y	
ISOLVLR	Y	Y	
ISOLVLR	Y	Y	
ISOLVLR	Y	Y	
ISOLVLR	Y	Y	
bndexpopt (bind explain option)	Y	Y	N
pkgownid (package owner identifier)	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
dgriopr1 (degree of I/O parallelism)	Y	Y	N

**Note:** <sup>1</sup> These options are supported only by DRDA level 4 and above.

### Command data objects

The REBIND command has the following command data object. (The list under the command data object shows the generic bind options that DB2 supports (sends and receives). Allowed values are shown in parentheses, if applicable.)

Table 78. REBIND command data object

REBIND command data object	Requester	Server	Required?
bndopt (bind option) <sup>1</sup>	S	Y	N
DBPROTOCOL(DRDA   PRIVATE) <sup>2</sup>	Y	Y	
DEFER(PREPARE) <sup>2</sup>	Y	Y	
NODEFER(PREPARE) <sup>2</sup>	Y	Y	
ENCODING(ASCII   EBCDIC   UNICODE   <i>ccsid</i> ) <sup>2</sup>	Y	Y	
FUNCPATH( <i>'schema-name'</i> )	N	Y	
IMMEDWRITE(NO   YES   PH1) <sup>2</sup>	Y	Y	
KEEPDYNAMIC(NO   YES) <sup>2</sup>	N	Y	
LANGUAGE(COBOL   COBOL2   IBMCOBOL) <sup>3</sup>	Y	Y	
MINDIVSCALE( <i>'number-char'</i> )	Y	Y	
OPTHINT( <i>'hint-id'</i> ) <sup>2</sup>	Y	Y	
PATH( <i>'schema-name'</i> ) <sup>2</sup>	Y	Y	
PATHDEFAULT <sup>2</sup>	Y	Y	
PRECOMPCCSID( <i>'number-string'</i> ) <sup>3</sup>	Y	Y	
REOPT(VARS) <sup>2</sup>	Y	Y	
NOREOPT(VARS) <sup>2</sup>	Y	Y	

**Notes:**

1. bndopt is supported by DRDA level 3 only.
2. See *DB2 Command Reference* for complete information about these bind options.
3. These options flow only between z/OS requesters and z/OS servers.

## Reply data objects

The REBIND command has the following reply data objects.

Table 79. REBIND reply data objects

REBIND reply data object	Requester	Server	Required?
typdefnam (data type definition name)	Y	N	N
typdefovr (data type definition override)	Y	N	N
sqlcard (SQLCA reply data)	Y	Y	Y
mgrlvlovr (manager level overrides)	Y	N	N

## Reply messages

The RDBUPDRM (RDB update) reply message has the following instance variables, which are relevant for DRDA level 2 only.

Table 80. Instance variables of the RDBUPDRM reply message

RDBUPDRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
rdbnam (RDB name)	Y	Y	Y
svrdgn (server diagnostic information)	Y	Y	Y



---

## SECCHK command

The SECCHK (security check) command sends information to the target security manager to authenticate the user.

### Command parameters

The SECCHK command has the following instance variables.

Table 81. SECCHK command instance variables

SECCHK command instance variable	Requester	Server	Required?
secmgrnm (security manager name)	Y	Y	N
secmec (security mechanism) <sup>1</sup>	Y	Y	Y
password (password)	Y	Y	N
usrld (user ID at the target system)	Y	Y	N
rdbnam (RDB name)	Y	Y	N
sectkn (security token)	N	Y	N

**Note:** <sup>1</sup> Table 7 on page 12 lists the security mechanisms that DB2 supports.

### Command data objects

The SECCHK command has the following command data object.

Table 82. SECCHK command data object

SECCHK command data object	Requester	Server	Required?
sectkn (security token)	Y	Y	N

### Reply data objects

The SECCHK command has the following reply data object.

Table 83. SECCHK reply data object

SECCHK reply data object	Requester	Server	Required?
sectkn (security token)	Y	Y	N

### Reply messages

The SECCHKRM (security check completed) reply message has the following instance variables.

Table 84. Instance variables of the SECCHKRM reply message

SECCHKRM reply message instance variable	Requester	Server	Required?
svrcod (severity code)	Y	Y	Y
secchkcd (security check code)	Y	Y	Y
svcerno (security service error number)	Y	Y	N
svdgn (server diagnostic information)	Y	Y	Y

---

## SNDPKT command

The SNDPKT (send packet) command tests connectivity between the source server and the target server.

As a requester, DB2 never sends the SNDPKT command.

## Command parameters

The SNDPKT command has the following instance variables.

*Table 85. SNDPKT command instance variables*

<b>SNDPKT command instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
respktsz (response packet size)	N/A	Y	N
rdbnam (RDB name)	N/A	Y	N

## Command data objects

The SNDPKT command has the following command data object.

*Table 86. SNDPKT command data object*

<b>SNDPKT command data object</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
pktobj (packet object)	N/A	Y	N

## Reply data objects

The SNDPKT command has the following reply data object.

*Table 87. SNDPKT reply data object*

<b>SNDPKT reply data object</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
pktobj (packet object)	N/A	Y	Y

## Reply messages

Normal processing of the SNDPKT command results in no DDM reply messages.

---

## SYNCCTL command

The SYNCCTL (sync point control) command conveys sync point information to the target.

### Command parameters

The SYNCCTL command has the following instance variables.

Table 88. SYNCCTL command instance variables

SYNCCTL command instance variable	Requester	Server	Required?
synctype (sync point operation type)	Y	Y	Y <sup>1</sup>
rlsconv (release conversation)	Y	Y	N <sup>2</sup>
uowid (unit of work identifier)	Y	Y	N <sup>3</sup>
forget (forget unit of work)	Y	Y	N <sup>3</sup>
xid (global transaction identifier)	Y	Y	N <sup>4</sup>
xidshr (share recoverable resources)	Y	Y	N <sup>3</sup>
xafflags (XA flags)	N	S	Y <sup>5</sup>
TMFAIL	N	Y	
TMJOIN	N	N	
TMLCS	N	N	
TMLLOCAL	N	Y	
TMNOFLAGS	N	Y	
TMONEPHASE	N	Y	
TMRECOVER	N	Y	
TMRESUME	N	N	
TMSUCCESS	N	Y	
TMSUSPENDS	N	N	
timeout (timeout)	N	Y	N <sup>6</sup>

**Notes:**

1. Required for both SYNCPTMGR and XAMGR.
2. Optional for both SYNCPTMGR and XAMGR.
3. Optional for SYNCPTMGR; not applicable for XAMGR.
4. Optional for SYNCPTMGR; required for XAMGR.
5. Not applicable for SYNCPTMGR; required for XAMGR.
6. Not applicable for SYNCPTMGR; optional for XAMGR.

### Command data objects

The SYNCCTL command has the following command data objects.

Table 89. SYNCCTL command data objects

SYNCCTL command data object	Requester	Server	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

Table 89. SYNCCTL command data objects (continued)

SYNCCTL command data object	Requester	Server	Required?
tcphost (fully qualified host domain name)	Y	Y	N
cnntkn (connection token)	Y	Y	Y
synclog (sync point log)	Y	Y	N <sup>1</sup>
<b>Note:</b> <sup>1</sup> Required for SYNCPTMGR when associated with particular <i>synctype</i> values; optional for XAMGR			

## Reply data objects

The SYNCCTL command has the following reply data objects.

Table 90. SYNCCTL reply data objects

SYNCCTL reply data object	Requester	Server	Required?
synctype (sync point operation type)	Y	Y	Y <sup>1</sup>
synclog (sync point log)	Y	Y	N <sup>2</sup>
sqlstt (SQL statements)	N	Y	N <sup>3</sup>
syncrcd (sync point control reply)	Y	Y	Y <sup>4</sup>
xaretvai (XA return value)	Y	Y	Y <sup>5</sup>
prphrcist (list of prepared and heuristically completed XIDs)	N	Y	N <sup>6</sup>

**Notes:**

1. Required for SYNCPTMGR; not applicable for XAMGR.
2. Optional for SYNCPTMGR; not applicable for XAMGR.
3. Optional for both SYNCPTMGR and XAMGR
4. Optional for SYNCPTMGR; required for XAMGR.
5. Not applicable for SYNCPTMGR; required for XAMGR.
6. Not applicable for SYNCPTMGR; optional for XAMGR.

## Reply messages

Normal processing of the SYNCCTL command, when *synctype* specifies a new unit of work, results in no DDM reply messages.

---

## SYNCRSY command

The SYNCRSY (sync point resync) command resolves indoubt units of work between sync point managers.

### Command parameters

The SYNCRSY command has the following instance variables.

*Table 91. SYNCRSY command instance variables*

<b>SYNCRSY command instance variable</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
rsynctyp (resync operation type)	Y	Y	Y
uowid (unit of work identifier)	Y	Y	N
uowstate (unit of work state)	Y	Y	N

### Command data objects

The SYNCRSY command has the following command data object.

*Table 92. SYNCRSY command data object*

<b>SYNCRSY command data object</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
synclog (sync point log)	Y	Y	N

### Reply data objects

The SYNCRSY command has the following reply data objects.

*Table 93. SYNCRSY reply data objects*

<b>SYNCRSY reply data object</b>	<b>Requester</b>	<b>Server</b>	<b>Required?</b>
rsynctyp (resync operation type)	Y	Y	Y
uowid (UOW identifier)	Y	Y	Y
uowstate (UOW state)	Y	Y	Y

### Reply messages

Normal processing of the SYNCRSY command results in no DDM reply messages.



---

## Chapter 2. Accounting for distributed data

Many businesses and institutions implement resource monitoring practices that allow system administrators to associate resource usage with individual user access. The results of these practices can be used, for example, to charge individual users or their departments for the resources they consume.

To enable an accounting or monitoring system to track DRDA access to a DB2 database server, a requester can send accounting and monitoring data to DB2 in one of two ways:

- Send an accounting identifier string in the *prddta* instance variable of the ACCRDB command with each application's connect request.  
The format of the string in *prddta* is described in "Format of prddta data."
- Send accounting or monitoring identifier strings in the *sqlstt* command data object of the EXCSQLSET command.

This is not limited to connect requests.

The format of the string in *sqlstt* is described in "Format of sqlstt data" on page 56.

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 DSN810.SDSNMACS.

---

### Format of prddta data

The *prddta* instance variable must contain the total length of the data, a code point, and one or more sets of data in the following format:

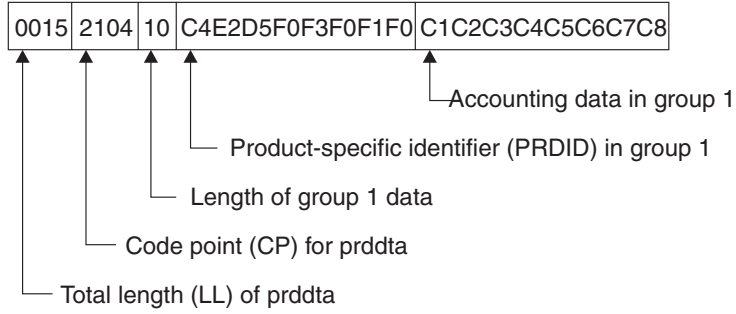
L        Length of the data that follows. The length value can be zero if no data follows for this set.

Data    Data of length L.

DB2 recognizes the first set of data as accounting data when the length of all sets of data equals the total length of *prddta*. If you send accounting data, you must always send it as the first set of data in the *prddta* instance variable.

Figure 2 on page 54 shows an example of one set of data in the *prddta* instance variable. All length values in the total length calculation are expressed in hexadecimal format.

The prddta data is:



The length of 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

---

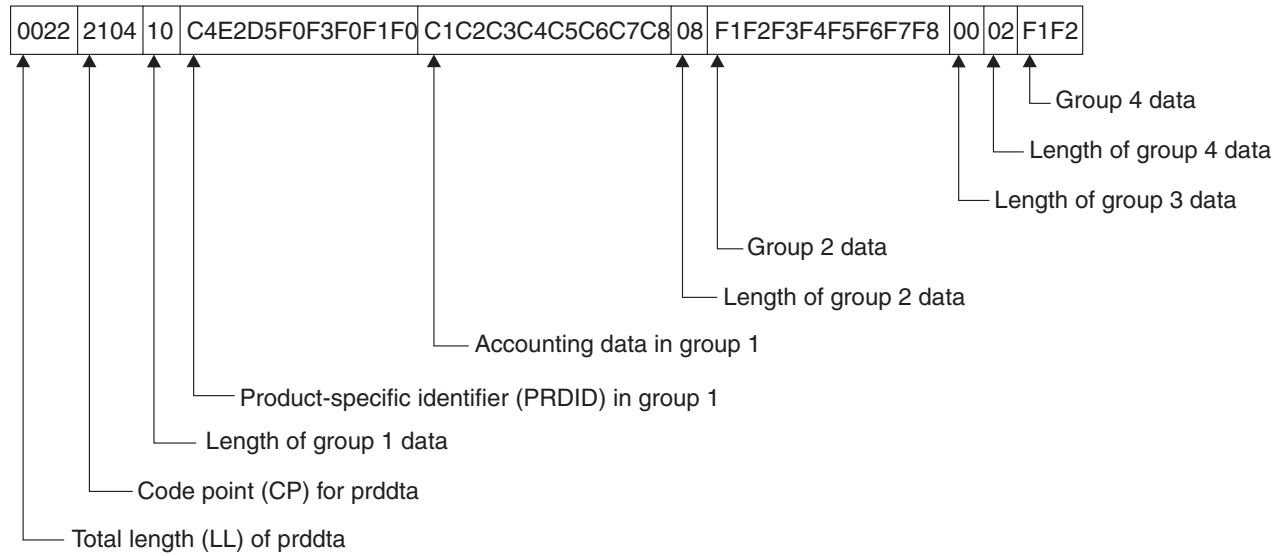
15 Total length of prddta

Figure 2. Example of one set of data in prddta

Figure 3 on page 55 shows an example of four sets of data in the *prddta* instance variable. All length values in the total length calculation are expressed in hexadecimal format.



The prddta data is:



The length of 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

Figure 3. Example of four sets of data in prddta

---

## Format of sqlstt data

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 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 ACCRDB command can be accepted, 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 DSNV4371 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 with one or more *sqlstt* command data objects. The information sent in the *sqlstt* command data object determines its format as follows:

- End user ID

To set the end user ID, *sqlstt* contains the string SET CLIENT USERID, followed by the user ID in single quotation marks. DB2 accepts a user ID of up to 16 characters and truncates any characters that exceed that length. For example, to set the end user ID to 'my\_eu\_userid':

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

- End user workstation name

To set the end user workstation name, *sqlstt* contains the string SET CLIENT WRKSTNNAME, followed by the workstation name in single quotation marks. DB2 accepts a name of up to 18 characters and truncates any characters that exceed that length. For example, to set the end user workstation name to 'my\_eu\_wstationname':

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

- End user application name

To set the end user application name, *sqlstt* contains the string SET CLIENT APPLNAME, followed by the application name in single quotation marks. DB2 accepts a name of up to 32 characters and truncates any characters that exceed that length. For example, to set the end user application name to 'my\_eu\_applname':

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

- Accounting information

To set the accounting information, *sqlstt* contains the string SET CLIENT ACCTNG, followed by the accounting information in single quotation marks. 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, to set the accounting information to 'DSN05010my\_acctng\_info':

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

Use a comma (,) to delimit string information. Represent hexadecimal string values as 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'
```



---

## Chapter 3. Distributing work across data sharing members in an SNA network

In an SNA environment, DB2 uses the DB2 Sysplex transaction program, an LU 6.2 Transaction Program Name (TPN), to allow DRDA requesters to determine which members of a DB2 data sharing group are currently active. The TPN returns a list of NETID.LUNAME values for the active DB2 servers in the group. The list also contains a weight for each server on the list. The weight reflects the current, available capacity of the server. This information allows requesters to distribute future requests for DB2 threads across the available members of a data sharing group on the basis of capacity.

The LU 6.2 parameters and messages for invoking the DB2 Sysplex transaction program are described in the rest of this chapter.

---

### Allocating a conversation to the Sysplex transaction program

The parameters that are used to allocate an LU 6.2 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 that the requester wants to receive the NETID.LUNAMES of the members in the DB2 data sharing 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 $22+(n \times 18)$ , 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 that this is a reply message containing a list of DB2 server NETID.LUNAME values.
4	18	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	$n \times 18$	$n$ occurrences of the following:

**NETID** VTAM<sup>®</sup> network name of the DB2 server group. This name is eight characters, padded on the right with blanks.

**LUNAME**

VTAM LU name of the DB2 server group. This name is eight characters, padded on the right with blanks.

**Weight** A 2-byte integer containing the weighting factor for the server that is identified by NETID.LUNAME. This number controls the proportion of LU 6.2 conversations that are directed to the server that is 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 that the value of  $n$  is 2. Weight 1 is 4. Weight 2 is 1.

With these values, 80% of the LU 6.2 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.

---

## 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 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
J46A/G4

555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document 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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Programming interface information

This book is intended to help you to write programs that communicate with IBM DB2 Universal Database for z/OS (DB2 UDB for z/OS) by means of DDM commands.

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

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

---

## Trademarks

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

CICS	ibm.com
DB2	IMS
DB2 Universal Database	OS/390
Distributed Relational Database Architecture	RACF
DRDA	VTAM
IBM	z/OS

UNIX is a registered trademark of The Open Group in the United States and other countries.



---

# Readers' Comments — We'd Like to Hear from You

DB2 Universal Database for z/OS  
Reference for Remote DRDA  
Requesters and Servers  
Version 8

Publication No. SC18-7424-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

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

---

Name

---

Address

---

Company or Organization

---

Phone No.



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  
H150/090  
555 Bailey Avenue  
San Jose, CA 95141-9989  
U. S. A.



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5625-DB2

Printed in USA

SC18-7424-00

