

Cincom

SUPRA SERVER PDM

RDM COBOL Programmer's Quick Reference
(OS/390 & VSE)

P26-8370-63



SUPRA[®] Server PDM RDM COBOL Programmer's Quick Reference (OS/390 & VSE)

Publication Number P26-8370-63

© 1986, 1988, 1994, 1996, 1998, 2000, 2002 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage [®]	iD CinDoc [™]	MANTIS [®]
C+A-RE [™]	iD CinDoc Web [™]	Socrates [®]
CINCOM [®]	iD Consulting [™]	Socrates [®] XML
Cincom Encompass [®]	iD Correspondence [™]	SPECTRA [™]
Cincom Smalltalk [™]	iD Correspondence Express [™]	SUPRA [®]
Cincom SupportWeb [®]	iD Environment [™]	SUPRA [®] Server
CINCOM SYSTEMS [®]	iD Solutions [™]	Visual Smalltalk [®]
	intelligent Document Solutions [™]	VisualWorks [®]
gOOj [™]	Intermax [™]	

UniSQL[™] is a trademark of UniSQL, Inc.
ObjectStudio[®] is a registered trademark of CinMark Systems, Inc.

All other trademarks are trademarks or registered trademarks of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U.S.A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: <http://www.cincom.com>

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *SUPRA Server PDM RDM COBOL Programmer's Quick Reference (OS/390 & VSE)*, P26-8370-63, is dated January 15, 2002. This document supports Release 2.7 of SUPRA Server PDM in IBM mainframe environments.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the [survey](#) provided with the online documentation at your convenience.

Cincom Technical Support for SUPRA Server PDM

FAX: (513) 612-2000
Attn: SUPRA Server Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: SUPRA Server Support
55 Merchant Street
Cincinnati, OH 45246-3732
U.S.A.



Contents

Conventions	vii
COBOL divisions	9
Identification division	9
Environment division	9
Data division.....	9
Procedure division.....	9
Program statements.....	10
INCLUDE statements in data division	10
Procedure division statements	11
DBAID commands.....	14
= command	14
BYE command	14
BY-LEVEL command	14
CAUTIOUS command.....	14
COLUMN-TEXT command	14
COMMIT command.....	15
DELETE command	15
ERASE command	15
FIELD-DEFN command	15
FORGET command	15
GET command.....	16
GO command.....	16
INSERT command	17
KEEP command.....	17
LINESIZE command	17
MARK command	17
MARKS command.....	17
OPEN command	18
PAGESIZE command	18
RELEASE command.....	18
RESET command	18
SIGN-OFF command.....	18
SIGN-ON command.....	19
SURE command	19
UPDATE command.....	19
USER-LIST command	19
USERS command.....	19

- VIEW-DEFN command 20
- VIEWS command 20
- VIEWS-FOR-USER command 20
- DBAID considerations..... 20
- Status indicators 21
 - ASI values..... 21
 - FSI values..... 21
 - VSI values..... 22
- Programming considerations 22
- Compiling..... 23
- Linking 24

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example
Constant width type	Represents screen images and segments of code.	<pre>PUT 'customer.dat' GET 'miller\customer.dat' PUT '\DEV\RMT0'</pre>
Slashed b (b)	Indicates a space (blank). The example indicates that four spaces appear between the keywords.	<pre>BEGINbbbSERIAL</pre>
Brackets []	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations: A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a WHERE clause. Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter either WAIT or NOWAIT. (WAIT is underlined to signify that it is the default.)	<pre>[WHERE <i>search-condition</i>]</pre> <pre><u>(WAIT)</u> (NOWAIT)</pre>
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter ON or OFF when using the MONITOR statement.	<pre>MONITOR {ON } {OFF}</pre>

Convention	Description	Example
<p><u>Underlining</u> (In syntax)</p>	<p>Indicates the default value supplied when you omit a parameter. The example indicates that if you do not choose a parameter, the system defaults to WAIT.</p>	<p>(WAIT) (NOWAIT)</p>
	<p>Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either STAT or STATISTICS.</p>	<p><u>STATISTICS</u></p>
<p>Ellipsis points...</p>	<p>Indicate that the preceding item can be repeated. The example indicates that you can enter multiple host variables and associated indicator variables.</p>	<p>INTO :host-variable [:ind-variable],...</p>
<p>UPPERCASE lowercase</p>	<p>In most operating environments, keywords are not case-sensitive, and they are represented in uppercase. You can enter them in either uppercase or lowercase.</p>	<p>COPY MY_DATA.SEQ HOLD_DATA.SEQ</p>
<p><i>Italics</i></p>	<p>Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you must substitute the name of a table.</p>	<p>FROM <i>table-name</i></p>
<p>Punctuation marks</p>	<p>Indicate required syntax that you must code exactly as presented.</p> <ul style="list-style-type: none"> () parentheses . period , comma : colon ' ' single quotation marks 	<p>(<i>user-id, password, db-name</i>) INFILE 'Cust.Memo' CONTROL LEN4</p>
<p>SMALL CAPS</p>	<p>Represent a required keystroke. Multiple keystrokes are hyphenated.</p>	<p>ALT-TAB</p>
<p>OS/390 VSE</p>	<p>Information specific to a certain operating system is flagged by a symbol in a shadowed box (OS/390) indicating which operating system is being discussed. Skip any information that does not pertain to your environment.</p>	<p>OS/390 See the SUPRA Server procedure library member TIS\$RDM for a list of RDM procedures. VSE See the SUPRA Server RDM sublibrary member TXJ\$INDX for a list of JCL.</p>

COBOL divisions

Identification division

IDENTIFICATION-DIVISION.

PROGRAM-ID. *program name*.

Environment division

ENVIRONMENT-DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

Data division

DATA-DIVISION.

FILE-SECTION.

WORKING-STORAGE SECTION.

LINKAGE SECTION.

Procedure division

Procedure Division statements.

Program statements

INCLUDE statements in data division

INCLUDE view-data statement

Indicates which views are needed by the program and their placement.

level-number[*user-view-name*]

INCLUDE *view-name* [(*user-column-list*)].

INCLUDE TIS-CONTROL statement

Must be coded in every program that issues an RDML request.

level-number **INCLUDE TIS-CONTROL.**

Code an INCLUDE statement in the LINKAGE SECTION to pass words to external modules (subroutines).

Use TIS-OPTIONS to specify DEBUG and TRACE.

Procedure division statements

COMMIT statement

Identifies the recovery point of the preceding task.

COMMIT.

DELETE statement

Removes a row from the database.

DELETE [ALL] *view-name*.

FORGET statement

Frees storage allocated by a previous MARK statement.

**FORGET *data-item* [NOT FOUND *cobol-imperative-statement*]
[ELSE *cobol-imperative-statement*].**

GET statement

Identifies the row to be retrieved from the view.

GET

NEXT
LAST
SAME
FIRST
PRIOR

view-name [FOR UPDATE] USING

<i>data-item</i> ₁ [... <i>data-item</i> ₉]
AT <i>data - item</i>

[NOT FOUND *cobol - imperative - statement*]
[ELSE *cobol - imperative - statement*]

INSERT statement

Inserts a new row into the view.

INSERT

NEXT
LAST
FIRST
PRIOR

view-name [**DUP KEY** *cobol-imperative-statement*]

MARK statement

Records the current position of the view established by the previous GET, UPDATE or INSERT statement.

MARK *view-name* **AT** *data-item*.

RELEASE statement

Closes a specific view or all views that have been opened, and frees internal storage.

RELEASE [*view-name*].

RESET statement

Restarts the task at the most recent commit point if possible.

RESET.

SIGN-OFF statement

Signs the user off from RDM.

SIGN-OFF.

SIGN-ON statement

Identifies the user to RDM.

SIGN-ON *user-name* [*password*].

UPDATE statement

Updates column values in the database.

UPDATE *view-name*.

DBAID commands

Used to test views before production use.

= command

Reissues the previous RDML command in its exact form.

=

BYE command

Exits DBAID.

BYE

BY-LEVEL command

Displays the column names in a view by level of occurrence, starting with the 0 level, followed by level 1, etc.

BY-LEVEL [*view-name* [*column-number*]

CAUTIOUS command

Prohibits an automatic COMMIT.

CAUTIOUS

COLUMN-TEXT command

Displays short and long text for a column in a view.

COLUMN-TEXT [*view-name* [*column-name*]

COMMIT command

Makes all updates since the last COMMIT permanent in the database.

COMMIT

DELETE command

Removes a view-row from the database.

DELETE [ALL] *view-name*

ERASE command

Issues an RDM RESET if an "X" FSI is returned. This command is the opposite of KEEP.

ERASE

FIELD-DEFN command

Displays the full description of columns in a view.

FIELD-DEFN [*view-name* [*column-name*]]

FORGET command

Frees the storage allocated by a previously issued MARK command.

FORGET *mark-name*

GET command

Retrieves and displays a row for the indicated view.

```
GET [ NEXT
      LAST
      SAME view-name
      FIRST
      PRIOR
    ]

[FOR UPDATE]
[ [AT mark-name ] ]
[ [USING literal1(literal2...literaln) ] ]
```

GO command

Issues a penetration GET request followed by a series of sweeping GET requests and displays the rows in tabular format.

```
GO [ NEXT
      PRIOR ] view-name

[ START { NEXT
          LAST
          SAME
          FIRST
          PRIOR
          AT mark-name } ]

FOR number-of-rows

[ { FROM } literal1(literal2...literaln)
  USING ]
```

INSERT command

Places a view row in the physical database at the relative location specified.

```
INSERT [ NEXT  
       LAST  
       FIRST  
       PRIOR ] view-name [MASS]
```

KEEP command

Prohibits an automatic RESET.

```
KEEP
```

LINESIZE command

Specifies the number of characters to be displayed in a line.

```
LINESIZE [number-of-characters]
```

MARK command

Marks the current position of the view row established by the previous GET command.

```
MARK view-name AT mark-name
```

MARKS command

Lists all open MARKs and the views they are marking.

```
MARKS
```

OPEN command

Readies a saved or virtual view for use by DBAID.

OPEN [*user-view-name*=]*view-name*[*column1*[,*...*],*columnn*]]

PAGESIZE command

Specifies the number of lines to display on a screen/page.

PAGESIZE [*number-of-lines*]

RELEASE command

Closes a specific view or all views that have been opened, and frees internal storage.

RELEASE [*view-name*]

RESET command

Rolls back any database updates since the most recent commit point.

RESET

SIGN-OFF command

Signs off the user from DBAID.

SIGN-OFF

SIGN-ON command

Identifies the user to DBAID.

SIGN-ON *user-name* [*password*]

SURE command

Causes a commit point after each successful INSERT, UPDATE or DELETE.

SURE

UPDATE command

Updates data values in the database.

UPDATE *view-name*

[*column*₁:=*literal*₁[,*...*,*column*_{*n*}:=*literal*_{*n*}]

USER-LIST command

Displays the column list for the specified user view.

USER-LIST *view-name*

USERS command

Displays information about the current users of the system.

USERS

VIEW-DEFN command

Displays a condensed description of a view.

VIEW-DEFN [*view-name*]

VIEWS command

Displays all of the views currently active in DBAID.

VIEWS

VIEWS-FOR-USER command

Lists the names and short text for the views related to the signed-on-user.

VIEWS-FOR-USER

DBAID considerations

* An asterisk has two uses:

- ◆ Substitutes for the last view name used.
- ◆ Denotes a comment line when entered in column 1.

Enter one command per line with a maximum of 72 characters.

Status indicators

ASI values

Value	Meaning
+	The column exists but has not changed since the last access. (GET processing only.)
=	The column exists but has changed since the last access. (GET processing only.)
-	The column is missing and has a null value. (GET processing only.)
V	The column contains an invalid value.
C	The column value has been changed by another view.
N	The column is set to its null value. (INSERT and UPDATE processing only.)

FSI values

Value	Meaning
*	Successful Completion
D	Data Error
F	Failure
N	Failure Due to Occurrence Problem
S	Security Check
U	Unavailable Resource
X	Reset Recommended

VSI values

Value	Meaning
C	The column value was changed by another view.
V	At least one invalid ASI was returned.
-	No invalid ASIs were returned, but at least one missing ASI was returned.
+	No invalid or missing ASIs were returned, but at least one physical occurrence in the database was returned.
=	No invalid, missing, or new physical occurrences were returned by this RDM function.

Programming considerations

- ◆ Each RDML statement must start on a line by itself and end with a period.
- ◆ Code an INCLUDE TIS-CONTROL statement in every program that issues an RDML request.
- ◆ CICS users must use command level COBOL. Macro level COBOL is not supported.
- ◆ Use the Relational Data Manager COBOL Programmer's Report for view layout, column names, and column type.
- ◆ Do not use COBOL reserved words in a view.

Compiling

When using VS/COBOL Compiler, use the APOST compile option (delimit strings with apostrophes). If this is not the default:

- ◆ **OS/390**—Specify DELIM=APOST in the JCL PARM options for the COBOL compile step.
- ◆ **VSE**—Specify CBL APOST as the first source statement in the preprocess input.



Also specify the NOSEQ option to avoid meaningless warning messages from the COBOL compiler.

Two parameters for precompiling:

- ◆ **Actual Compiler Name**—Required on parm of RDML Precompiler execution step.
 - **Batch**—Specify IKFCBLOO.
 - **CICS**—Specify CICS command level preprocessor name.
- ◆ **Miscellaneous Parameters**—Optional parameters for the COBOL Compiler or CICS preprocessor, or NOCOMPILE to bypass the COBOL compile.

Linking

OS/390

In batch OS/390, COBOL applications are linked with CSVILUV. CSILUV is a composite containing CSVILOAD and DATBAS.

In OS/390 CICS, COBOL applications are linked with CSVCLUV, CSVICICS, or CSVNICIC. If the application is to run above the 16 megabyte line, then the application must be linked with CSVNICIC.

VSE

In batch VSE, COBOL applications are linked with CSVIOSVS, a composite containing CSVJLUV and DATBAS.

In VSE CICS, COBOL applications are linked with SCVCLUV, CSVCICS, or CSVNICIC. If the application is to run above the 16 megabyte line or if the application allocates RDM parameters above the 16 megabyte line, then the application must be linked with CSVNICIC.
