



Datacom Bridge Services User's Guide

The contents of this manual may be revised without prior notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical for any purpose without the expressed written permission of LegacyJ Corporation.

©Copyright LegacyJ Corporation 2005
All Rights Reserved.

Fourth Edition: August , 2005

LegacyJ has made every effort to ensure that this manual is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time.

© Copyright LegacyJ Corporation 2005.
All Rights Reserved

Preface

The LegacyJ Datacom Bridge Services (LDBS) User's Guide provides guidelines for the usage of Datacom bridge interface with LegacyJ PERCObol. The use of Datacom Bridge Services with PERCObol and LegacyJ Transaction Platform provides access to data needed by applications migrated using LegacyJ technology.

Trademarks

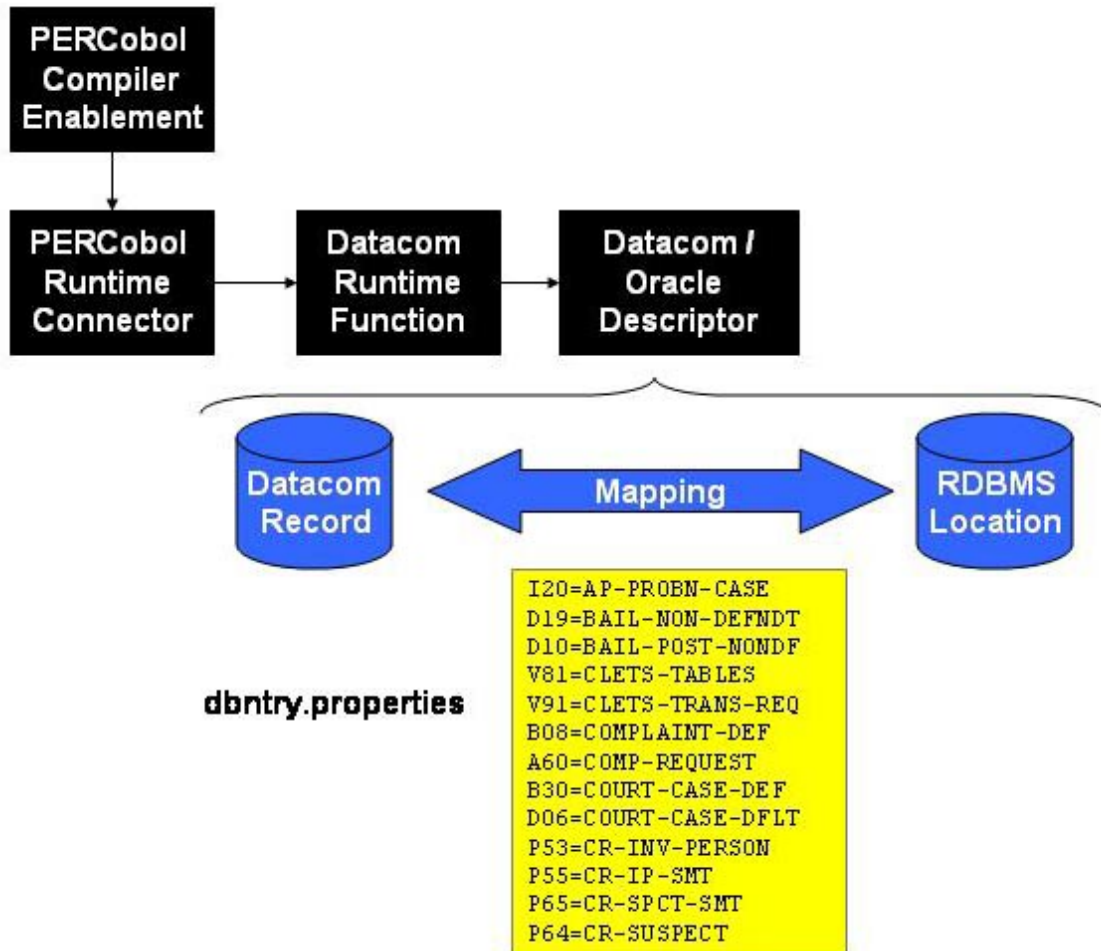
- ◆ PERCobol, LegacyJ Transaction Platform, ISAMJ, ViewJ and LegacyJ are trademarks of LegacyJ Corporation.
- ◆ AIX, zOS, zLinux and iSeries are trademarks of International Business Machines.
- ◆ HP-UX and MPE are registered trademarks of Hewlett-Packard Corporation
- ◆ IBM is the registered trademark of International Business Machines.
- ◆ Java is a trademark of Sun Microsystems
- ◆ UNIX is a registered trademark licensed exclusively to X/Open Company Limited.
- ◆ Windows XP and Windows Server are trademarks of Microsoft Corporation.
- ◆ Other company, product and service names may be trademarks of service marks of others.

Contents

BRIDGE FLOW	1
DATACOM DBNTRY FUNCTIONS.....	2
PROPERTIES MAPPING.....	6
DATABASE FIELD TO FIELD MAPPING.....	7
COMPOUND KEY MAPPING	8
COBOL CODE EXAMPLE	9
DATACOM BRIDGE ENABLEMENT MODULE	10
COMPILE REQUIREMENTS	11
RETURN CODES	12
INDEX.....	15

Bridge Flow

The LegacyJ Datacom Bridge Services provides the DBNTRY function as a replacement for using Datacom as the application database. LegacyJ Datacom Bridge works in conjunction with LegacyJ Transaction Suite and/or LegacyJ PERCOBOL Transaction Version. The following graphic illustrates the execution of the Datacom:



Datacom DBNtry Functions

Datacom function definitions are supported with the LegacyJ Bridge and function in a “like manner” they did in the legacy Datacom environment.

Functions	Call Name	Description
ABEND	ABEND	Abnormally terminate execution of a batch job.
ADD-RECORD	ADDIT	Insert a record into a specified table.
COUNT-RANGE	CNTR	Determine the number of records between two key values in two key areas. CNTR uses the two values as search boundaries and determines the number of records in a table which are equal to or between those boundaries. CNTR also determines the number of unique key values which exist in the same range.
COUNT-KEY	CNTKY	Determine the number of records in a table having a key value stored in the index which is equal to the value specified in the key area for the specified key name.
COUNT-FILE	CNTTB	Looks up the count information in a control block (within Datacom) does not actually scan the index.
COMMIT	COMIT	Commits a transaction creating a stable state for the transaction. The functions requiring a commit are now auto-committed and the COMIT is extra.
DELETE-RECORD	DELET	Delete a record from a table and deletes all keys associated with the record. DELET also releases exclusive control of the record. Before a record can be deleted, two conditions must be met. The record must be read with exclusive control, and the table specified in this command must have been defined as a table which may be updated.
GET-SEQUENTIAL	GETIT	Retrieves records sequentially from any table in a database, sequenced by any valid key name, starting with a key value greater than or equal to a specified key value. GETIT optionally obtains exclusive control of a record and either releases or retains exclusive control of the record previously read for update.
SET-TO-STARTING-KEY	GSETL	The table name, key name, and key value needed to establish a starting position for subsequent sequential processing. GSETL retrieves no data. It is a prerequisite command to GETIT.
LOCATE-LOWER	LOCBR	Locates the next lower record in the index. Synonym key values are returned in descending sequence by record ID.
LOCATE-KEY	LOCKG	Obtains the record ID of the first record in a table having a key value in the index greater than or equal to the key value requested.
LOCATE-KEY-INTERSECT	LOCKI	Enables the test of logical intersections of two keys without the need to read records.
LOCATE-EQ-LOWER	LOCKL	Locates the first record with a key value in the index equal to or lower than the key value specified in the request. The lower value is the highest value found lower than the specified value. When more than one entry exists for the specified key value, the entry with the highest record ID is retrieved.

Functions	Call Name	Description
LOCATE-KEY-RANGE	LOCKR	Provides the ability locate a record within a specified key value range. Specify the beginning and ending key values of the range in a 360-byte key area. LOCKR uses these two values as boundaries and locates the first record within the specified range.
LOCATE-KEY-EXACT	LOCKX	Locates the first record with a key in the index exactly equal to the key value presented in the request.
LOCATE-NEXT-EQUAL	LOCNE	Locates the next record with a key value equal to the specified key value.
LOCATE-NEXT-KEY	LOCNK	Locates the record with the next sequential key value in the index. The LOCNK function is opposite of LOCNE in that it returns only unique key values.
LOCATE-NEXT-RANGE	LOCNR	Enables the location of the next record with a key value within a specified range. After a LOCKR is issued and a record within a range is found, the LOCNR command enables you to find the key with the next greater value within that range.
LOCATE-NEXT	LOCNX	Locates the record in the next sequential entry in the index.
CHECKPOINT	COMIT	Commits a transaction creating a stable state for the transaction.
ADD-TO-LOG	LOGIT	Used to record checkpoint synchronization data.
READ-LOG	LOGLB	Used to retrieve records previously written by the task with a LOGIT or LOGDW command. The LOGIT or LOGDW data is returned in the same format as it was presented with the LOGIT or LOGDW command (data length, followed by the data).
BACKOUT-TRANS READ-BACKWARD	LOGTB	Used to reverse all transactions for a task which has been executed since the last OPEN, COMIT, CLOSE, LOGCR, or LOGCP record was written to the Log.
READ-BACKWARD-UPD	REDBR	Returns actual data elements from a database by reading the next record with the previous sequential entry in the index. REDBR does not obtain exclusive control.
	RDUBR	Returns actual data elements from a database by reading the next record with the previous sequential entry in the index. RDUBR acquires exclusive control for update purposes.
READ-BY-ID	REDID	Returns the actual data elements from a database by reading a specified record. The record is identified by specifying its record ID and the name of the table which contains it. REDID does not obtain exclusive control.
READ-BY-ID-UPD	RDUID	Returns the actual data elements from a database by reading a specified record. The record is identified by specifying its record ID and the name of the table which contains it. RDUID acquires exclusive control of the record for update purposes.
READ-GR-EQ	REDKG	Obtains the first record in a table that has a key value in the index greater than or equal to the key value requested. REDKG does not obtain exclusive control.

Functions	Call Name	Description
READ-GR-EQ-UPD	RDUKG	Obtains the first record in a table that has a key value in the index greater than or equal to the key value requested. RDUKG acquires exclusive control of the record for update purposes.
READ-LESS-EQ	REDKL	Obtain the first record in a table that has a key value in the index less than or equal to the key value specified in the request. REDKL does not obtain exclusive control.
READ-LESS-EQ-UPD	RDUKL	Obtain the first record in a table that has a key value in the index less than or equal to the key value specified in the request.
READ-RANGE	REDKR	Locates a record within a specified key value range. REDKR does not obtain exclusive control.
READ-RANGE-UPD	RDUKR	Locates a record within a specified key value range. RDUKR acquires exclusive control of the record for update purposes.
READ-KEY-EXACT	REDKX	Returns the actual data elements of a database by reading a specified record. Specify the record to be read by its key value and the name of the table containing the record. REDKX does not obtain exclusive control.
READ-KEY-EXACT-UPD	RDUKX	Returns the actual data elements of a database by reading a specified record. Specify the record to be read by its key value and the name of the table containing the record. RDUKX acquires exclusive control of the record for update purposes.
READ-LOC-ENTRY	REDLE	Returns the actual data elements from a database by reading a record previously located by one of the locate commands. REDLE does not obtain exclusive control.
READ-LOC-ENTRY-UPD	RDULE	Returns the actual data elements from a database by reading a record previously located by one of the locate commands. RDULE acquires exclusive control of the record for update purposes.
READ-NXT-EQ-KEY	REDNE	Returns actual data elements from a database by reading a record. REDNE reads the next record with a key value equal to the key value specified.
READ-NXT-EQ-KEY-UPD	RDUNE	Returns actual data elements from a database by reading a record. RDUNE reads for update the next record with an equal key value.
READ-NXT-KEY-VALUE	REDNK	Read (read for update) the record with the next sequential key value in the index. It performs the function opposite that of REDNE/RDUNE in that it only returns unique key values. REDNK does not obtain exclusive control.
READ-NXT-KEY-VALUE-UPD	RDUNK	Read (read for update) the record with the next sequential key value in the index. It performs the function opposite that of REDNE/RDUNE in that it only returns unique key values. RDUNK acquires exclusive control of the record for update purposes.
READ-NEXT-RANGE	REDNR	Locate the next record with a key value within a specified range. REDNR does not obtain exclusive control.
READ-NEXT-RANGE-UPD	RDUNR	Locate the next record with a key value within a specified range. RDUNR acquires exclusive control of the record for update purposes.

Functions	Call Name	Description
READ-NEXT	REDNX	Reads the record in the next sequential entry in the index.
READ-NEXT-UPD	RDUNX	Reads for update the record in the next sequential entry in the index.
RELEASE-RECORD	RELES	Releases exclusive control of a record. It does not write the record back to the table or destroy the condition of the Request Area. After the RELES command, select the next desired command and continue.
RELEASE-FILE	RELFL	Releases from exclusive control all records in a specified table held by the task.
UPDATE-RECORD	UPDAT	Modifies data elements and releases the exclusive control of the record. Elements not specified in the UPDAT command are not changed.

Properties Mapping

The LegacyJ Datacom Bridge Services data mapping currently is in a sequential file that contains the “from” and “migrated” data locations. The mapping is a simple mapping of the program data referenced to the actual data location as it is stored after it has been migrated from Datacom to the RDBMS (Oracle, DB2, etc.)

Database Field to Field Mapping

Data mapping between Datacom and the target database can be a simple one-to-one mapping with a descriptor defining the data type. Descriptor information is added to the mapping so that the appropriate formatting is passed back to the sending program. To properly define field mapping a "Description Block Header" and "Detail Descriptor" must be included in the Datacom Properties file.

General format

Format 1 (Description Block Header)

from-field=to-field

Format 2 (Detail Descriptor)

from-field.T=field type

from-field.LNGTH=field length

Syntax rules

The datacom properties description for a single field mapping (non-concatenated) is the combination of a Description Block Header and a Detail Descriptor or the combination of the Format 1 and Format 2 properties.

Format 1 (Description Block Header)

<i>from-field</i>	Datacom data location identifier. This will be the 3 character table id separated by a period and the 5 character field id (<i>example I20.PROBN</i>)
<i>to-field</i>	RDBMS data location identifier. This field must reside in a SQL database such as Oracle or DB2

Format 2 (Detail Descriptor)

<i>from-field .T=</i>	The field type definition which can either be C, N, B (<i>C=Character, N=Numeric, B=Binary</i>)
<i>from-field .LNGTH=</i>	The field length in numeric value.

Syntax Sample

```
I20 .PROBN=AP-PROBN-CASE  
I20 .PROBN .T=C  
I20 .PROBN .LNGTH=10
```

Compound Key Mapping

Compound or Conjunctive Mapping is a method for describing data materials comprised of multiple table entries. The following example illustrates a key made up of multiple parts which are not necessarily tightly associated.

The method for concatenating this RDBMS table entries is through the use of the ampersand "&".

General format

Format 3 (Conjunctive Field Format)

<i>from-field</i>	Datacom data location identifier. This will be the 3 character table id separated by a period and the 5 character field id (<i>example I20.PROBN</i>)
<i>to-field</i>	RDBMS data location identifiers. Each field to be combined must be separated with an ampersand (&). These fields must reside in a SQL database such as Oracle or DB2

Format 4 (Conjunctive Field Descriptor)

<i>from-field .tofield.T=</i>	The field type definition which can either be C, N, B (<i>C=Character, N=Numeric, B=Binary</i>)
<i>from-field .tofield.LNGTH=</i>	The field length in numeric value.

Syntax Sample

```
D12.WNTKY=&WNT-CATEGORY-CD&WARRANT-ID&DT-WNT-REC-CC&DT-WNT-REC-YR&DT-WNT-REC-MO&DT-WNT-REC-DA&TM-WNT-REC-HH&TM-WNT-REC-MM&SEQ-NO-WNT  
D12.WNTKY.WNT-CATEGORY-CD.T=C  
D12.WNTKY.WNT-CATEGORY-CD.LNGTH=8  
D12.WNTKY.WARRANT-ID.T=C  
D12.WNTKY.WARRANT-ID.LNGTH=12  
D12.WNTKY.DT-WNT-REC-CC.T=N  
D12.WNTKY.DT-WNT-REC-CC.LNGTH=6  
D12.WNTKY.DT_WNT-REC-YR.T=N  
D12.WNTKY.DT_WNT-REC-YR.LNGTH=4  
D12.WNTKY.DT-WNT-REC-MO.T=C  
D12.WNTKY.DT-WNT-REC-MO.LNGTH=3  
D12.WNTKY.DT-WNT-DA.T=N  
D12.WNTKY.DT-WNT-DA.LNGTH=2  
D12.TM-WNT-REC-HH.T=N  
D12.TM-WNT-REC-HH.LNGTH=2  
D12.TM-WNT-REC-MM.T=N  
D12.TM-WNT-REC-MM.LNGTH=2  
D12.SEQ-NO-WNT.T=N  
D12.SEQ-NO-WNT.LNGTH=6
```

COBOL Code Example

The following code sample illustrates how Datacom functions are called in COBOL.

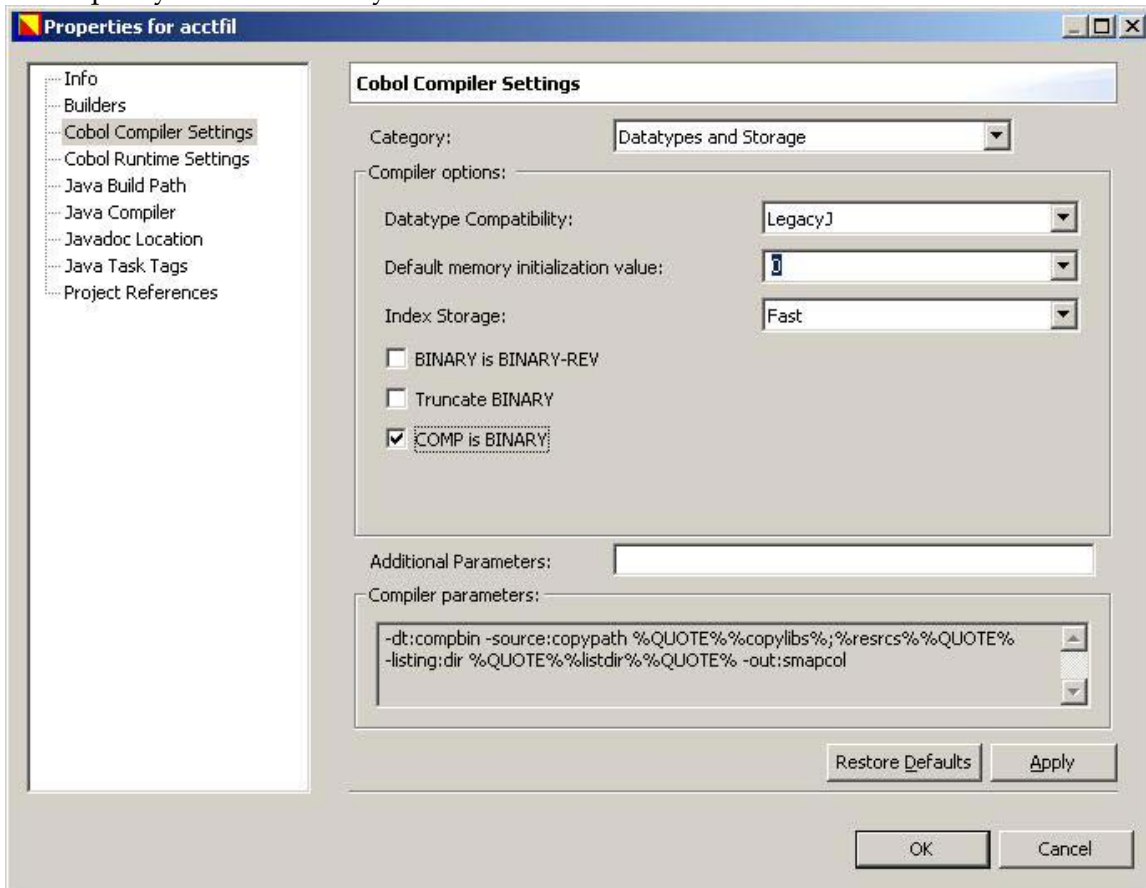
```
CALL 'DBENTRY' USING WF-APPLICATION-ID  
WF-A12-FIL-RQ-AREA  
WF-A12-FIL-WORK-AREA  
WF-A12-FIL-ELM-LIST.
```

Datacom Bridge Enablement Module

The LegacyJ Datacom Bridge enablement module functions in conjunction with the LegacyJ PERCobol Runtimes and must be present in order for DBNTRY calls within a COBOL program to be caught and handled by the LegacyJ Datacom Bridge. The DBNTRY class file (Java executable) must be placed in the COBOL project as a Java resource and must be deployed along with the Datacom application Integrated Development Environment project so that it is used (referenced) whenever a DBNTRY call is invoked.

Compile Requirements

Applications using Datacom support must be compiled with the LegacyJ PERCobol Transaction Version. The compile parameters for all modules requiring Datacom support must specify COMP is Binary.



RETURN CODES

The following return codes may be generated by the LegacyJ Datacom Bridge (DBNTRY):

Return Code	Commands	Message
bb	Successful operation	
spaces	Successful operation	
01	Unsupported Command	Unknown Command
02	CLOSE, OPEN	INVALID TABLE NAME The table name cannot be found in the Directory. (Not in Properties Table)
03	CNTxx, GETIT, GSETL, LOCxx, RDUNE, RDUNX, REDNE, REDNX	INVALID KEY NAME An invalid key name has been detected.
04	REDLE, REDID, REDLE, RDULE	INVALID RECORD ID The record ID does not point to a record.
05	ADDIT, CNTxx, GSETL, GSETP, LOCxx, OPEN, RDUxx, REDxx, RELxx, SELFR, SELNR, SELPR, SELSM	TABLE NOT OPEN The requested table was not opened for processing.
06	ADDIT, SELFR, SELNR, SELSM, UPDAT	TABLE NOT OPEN FOR UPDATE Tried to read a table with update intent or to modify a table for which there was no update authorization.
07	ADDIT	DATA AREA FULL A data area is full.
08	ADDIT	SYSTEM AREA FULL A system area is full.
09	DELET, GETIT, GETPS, LOCBR, LOCNX, RDULE, RDUNE, RDUNX, REDLE, REDNE, REDNX, SELNR, SELSM, UPDAT	PREREQUISITE COMMAND ERROR A previous command or command sequence or an alteration of the Request Area indicates the current request is not valid. <i>Rowid is null or spaces; the rowid is invalid.</i>
10	ADDIT	DUPLICATE MASTER KEY NOT ALLOWED Unique constraint violated; operation tried to insert a record with a duplicate key.
11	UPDAT	MASTER KEY HAS BEEN MODIFIED – UPDATE REJECTED An update request to change a Master Key has been rejected.
12	RDUID, RDULE, REDID, REDLE, SELSM	SPECIAL DELETED RECORD A read command could not be processed.

Return Code	Commands	Message
13	All Commands	Internal Bridge Error
14	LOCxx, RDUID, RDUKX, RDUNE, RDUNX, REDID, REDKX, REDNE, REDNX, SELFR, SELNR	NO RECORD FOUND Record not found matching the request requirements.
16	GETIT, RDUxx, REDxx, SELFR, SELNR, SELSM	EXCLUSIVE CONTROL INTERLOCK The record you requested for update is being held under exclusive control by another task.
18	REDxx, RDUxx, SELNR, SELSM	EXCLUSIVE CONTROL DUPLICATE There has been a duplication of exclusive control error.
19	GETIT, GETPS	END OF TABLE FOR GETIT, END OF AREA FOR GETPS An error occurred during sequential processing.
22	ADDIT, DELET, GETIT, GSETP, RDUxx, REDxx, SELFR	ELEMENT NAME NOT FOUND Element on the Element List is not valid for the table. SQL cannot find identifier.
23	ADDIT, DELET, GETIT, GSETP, RDUxx, REDxx, SELFR	ELEMENT SECURITY CODE VIOLATION A security-protected element was requested, but the security code in the Element List did not match.
25	OPEN	INVALID DATABASE ID The database ID is not valid.
29	LOGLB	EOF DURING LOGLB COMMAND There are no more LOGIT records to retrieve. Element name not found; invalid identifier.
36	ADDIT, CNTx, GSETL, GSETP, LOCxx, OPEN, RDUxx, REDxx, RELxx, SELFR, SELNR, SELPR, SELSM	USER VIEW NOT OPEN The request was made for an unopened database.
76	All Commands	Failure to obtain a SQL Connection
90	ADDIT	SQL ERROR -- Unique Constraint Violation
91	All Commands	SQL Table not found
92	SEFR	SET SELECTION INTERRUPT An interrupt has occurred during the set selection process of the SELFR command. The number of records selected at the point of interrupt is in the SET-RECORD-COUNT field of the Request Area.
93	SELNR	ATTEMPT TO POSITION PAST END/BEG-OF-SET The skip-count value is greater than the number of records remaining.

Return Code	Commands	Message
94	COMIT, ROLBK	GENERAL ERROR INDICATOR
95	Returned by all supported commands.	No connection available to SQL database.
96	CNTxx	Incorrect Count value generated
97	CNTxx	No results generated from count
98	Returned by all supported commands	Miscellaneous SQL error
99	All commands	Bridge Error

Index

Compound, 8
Compound Key Mapping, 8
Conjunctive Field Descriptor, 8
Conjunctive Field Format, 8
Data mapping, 7
Datacom, i, v, 1, 2, 6, 7, 8, 9, 10, 11, 12
DBNTRY, 1, 9, 10, 12
Description Block Header, 7
Detail Descriptor, 7
HP-UX, vii
IBM, vii
Java, vii
LegacyJ Datacom Bridge, 1, 10
Microsoft, vii
MPE, vii
PERCobol, vii
PERCobol Runtimes, 10
RDBMS, 6, 7, 8
Sun Microsystems, vii
UNIX, vii
Windows NT, vii