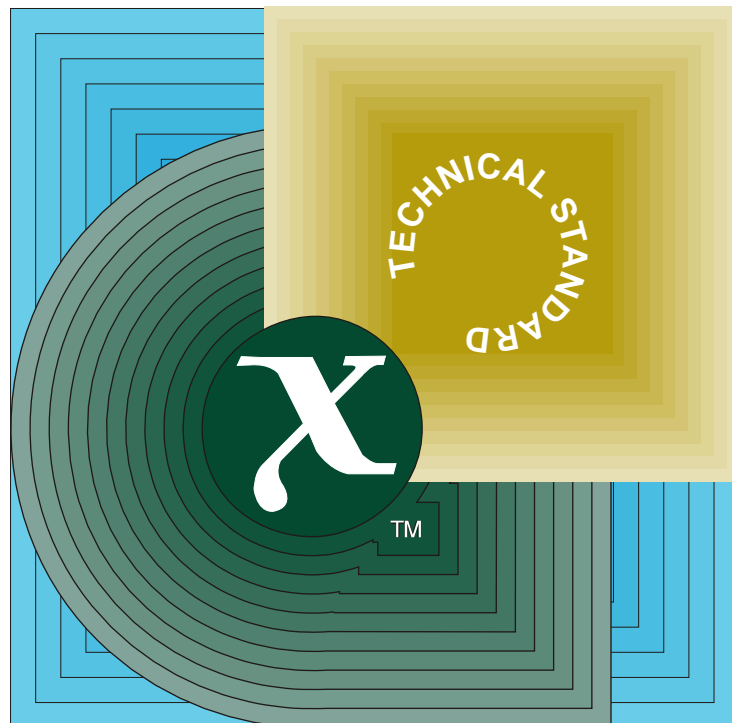


Technical Standard

Data Management: SQL Remote Database Access



THE *Open* GROUP

[This page intentionally left blank]

X/Open CAE Specification

Data Management:

SQL Remote Database Access

X/Open Company Ltd.



© July 1993, X/Open Company Limited

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

X/Open CAE Specification

Data Management: SQL Remote Database Access

ISBN: 1-872630-98-7

X/Open Document Number: C307

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.co.uk

Contents

Chapter 1	Introduction.....	1
Chapter 2	Changes to the Referenced Documents	3
2.1	Typographical Equivalence	3
2.2	sQLDBLStatementNotAllowed Error	4
2.3	SQL Diagnostics Information.....	5
2.4	VARCHAR Data Type	8
2.5	Dynamic SQL Statements	10
2.6	Connection Management Statements.....	11
2.7	Data Definition Statements.....	12
Chapter 3	Implementation Agreements.....	13
3.1	Character Set.....	13
3.2	Operation Limits.....	13
3.3	Transactions	13
3.4	RDA Features Not Used in This Specification.....	13
3.5	Protocol Stack	14
3.5.1	Session Layer	14
3.5.2	Presentation Layer.....	14
3.5.3	Association Control Service Element (ACSE).....	14
3.6	RDA Parameters.....	15
3.6.1	R-Initialize Service.....	16
3.6.2	R-Synchronize APDU	18
3.6.3	R-Terminate Service	18
3.6.4	R-BeginTransaction Service	19
3.6.5	R-Commit Service.....	20
3.6.6	R-Rollback Service.....	21
3.6.7	R-Cancel Service	22
3.6.8	R-Status Service.....	23
3.6.9	R-Open Service.....	25
3.6.10	R-Close Service.....	27
3.6.11	R-ExecuteDBL Service	28
3.7	Limits for Common Parameters	30
3.8	Object Identifiers.....	33
3.8.1	Object Identifiers Defined By X/Open.....	33
3.8.2	Object Identifiers Defined By Other Authorities.....	33
3.9	Prospective Uses of Parameters.....	34
Appendix A	Resulting Text of Table 10.....	35

Appendix B	ASN.1 Module with X/Open Changes	39
	Glossary	59
	Index.....	61

List of Tables

3-1	Parameters for R-Initialize request.....	16
3-2	Parameters for R-Initialize result response	17
3-3	Parameters for R-Initialize error response.....	17
3-4	Parameters for R-Terminate request.....	18
3-5	Parameters for R-Terminate result response.....	18
3-6	Parameters for R-Terminate error response	18
3-7	Parameters for R-BeginTransaction request.....	19
3-8	Parameters for R-BeginTransaction error response	19
3-9	Parameters for R-Commit request	20
3-10	Parameters for R-Commit result response	20
3-11	Parameters for R-Commit error response.....	20
3-12	Parameters for R-Rollback request.....	21
3-13	Parameters for R-Rollback result response	21
3-14	Parameters for R-Rollback error response.....	21
3-15	Parameters for R-Cancel request	22
3-16	Parameters for R-Cancel result response.....	22
3-17	Parameters for R-Cancel error response	22
3-18	Parameters for R-Status request	23
3-19	Parameters for R-Status result response	23
3-20	Parameters for R-Status error response	24
3-21	Parameters for R-Open request.....	25
3-22	Parameters for R-Open result response	25
3-23	Parameters for R-Open error response.....	26
3-24	Parameters for R-Close request	27
3-25	Parameters for R-Close result response	27
3-26	Parameters for R-Close error response.....	27
3-27	Parameters for R-ExecuteDBL request.....	28
3-28	Parameters for R-ExecuteDBL result response.....	28
3-29	Parameters for R-ExecuteDBL error response	29
3-30	Parameters for SQLDataTypeDescriptor	30
3-31	Parameters for SQLDBLException.....	31
3-32	Parameters for SQLValue	32

Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and allows users to move between systems with a minimum of retraining.

The components of the Common Applications Environment are defined in X/Open CAE Specifications. These contain, among other things, an evolving portfolio of practical application programming interfaces (APIs), which significantly enhance portability of application programs at the source code level, and definitions of, and references to, protocols and protocol profiles, which significantly enhance the interoperability of applications.

The X/Open CAE Specifications are supported by an extensive set of conformance tests and a distinct X/Open trademark - the XPG brand - that is licensed by X/Open and may be carried only on products that comply with the X/Open CAE Specifications.

The XPG brand, when associated with a vendor's product, communicates clearly and unambiguously to a procurer that the software bearing the brand correctly implements the corresponding X/Open CAE Specifications. Users specifying XPG-conformance in their procurements are therefore certain that the branded products they buy conform to the CAE Specifications.

X/Open is primarily concerned with the selection and adoption of standards. The policy is to use formal approved *de jure* standards, where they exist, and to adopt widely supported *de facto* standards in other cases.

Where formal standards do not exist, it is X/Open policy to work closely with standards development organisations to assist in the creation of formal standards covering the needed functions, and to make its own work freely available to such organisations. Additionally, X/Open has a commitment to align its definitions with formal approved standards.

X/Open Specifications

There are two types of X/Open specification:

- *CAE Specifications*

CAE (Common Applications Environment) Specifications are the long-life specifications that form the basis for conformant and branded X/Open systems. They are intended to be used widely within the industry for product development and procurement purposes.

Developers who base their products on a current CAE Specification can be sure that either the current specification or an upwards-compatible version of it will be referenced by a future XPG brand (if not referenced already), and that a variety of compatible, XPG-branded systems capable of hosting their products will be available, either immediately or in the near future.

CAE Specifications are not published to coincide with the launch of a particular XPG brand, but are published as soon as they are developed. By providing access to its specifications in this way, X/Open makes it possible for products that conform to the CAE (and hence are eligible for a future XPG brand) to be developed as soon as practicable, enhancing the value of the XPG brand as a procurement aid to users.

- *Preliminary Specifications*

These are specifications, usually addressing an emerging area of technology, and consequently not yet supported by a base of conformant product implementations, that are released in a controlled manner for the purpose of validation through practical implementation or prototyping. A Preliminary Specification is not a “draft” specification. Indeed, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE Specification.

Preliminary Specifications are analogous with the “trial-use” standards issued by formal standards organisations, and product development teams are intended to develop products on the basis of them. However, because of the nature of the technology that a Preliminary Specification is addressing, it is untried in practice and may therefore change before being published as a CAE Specification. In such a case the CAE Specification will be made as upwards-compatible as possible with the corresponding Preliminary Specification, but complete upwards-compatibility in all cases is not guaranteed.

In addition, X/Open periodically publishes:

- *Snapshots*

Snapshots are “draft” documents, which provide a mechanism for X/Open to disseminate information on its current direction and thinking to an interested audience, in advance of formal publication, with a view to soliciting feedback and comment.

A Snapshot represents the interim results of an X/Open technical activity. Although at the time of publication X/Open intends to progress the activity towards publication of an X/Open Preliminary or CAE Specification, X/Open is a consensus organisation, and makes no commitment regarding publication.

Similarly, a Snapshot does not represent any commitment by any X/Open member to make any specific products available.

X/Open Guides

X/Open Guides provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant.

X/Open Guides are not normative, and should not be referenced for purposes of specifying or claiming X/Open-conformance.

This Document

This document is a CAE Specification (see above). It relies on the ISO/IEC RDA SQL standard, which specifies a message format for remote communication of SQL database language statements (query and update) to a remote database. This specification defines uses of the message fields and other implementation information such as sequencing and optional features. It shows how SQL statements map to the Remote Database Access (RDA) protocol.

This document is the result of a joint collaborative effort between the X/Open Data Management Working Group and the SQL Access Group.

This specification is structured as follows:

- Chapter 1 is a brief introduction to RDA and to the need for the refinements that X/Open has specified.
- Chapter 2 specifies a set of formal changes to the referenced ISO documents to produce a version that is usable with X/Open SQL.
- Chapter 3 describes agreements at the protocol level on maximum values and on the use of parameters required for use with X/Open SQL.
- Appendix A shows the resulting text of Table 10 of ISO/IEC 9579-2 when the changes specified in Chapter 2 are applied.
- Appendix B shows the resulting ASN.1 module when the changes specified in Chapter 2 are applied.

Typographical Conventions

The following typographical conventions are used throughout this document:

- **Bold** font is used in text for filenames, keywords, type names, data structures and their members.
- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition. Italics in text also denote variables or substitutable items.
- Normal font is used for the names of constants and literals.
- Syntax and code examples are shown in `fixed width` font.
- Variables within syntax statements are shown in *italic fixed width* font.

Because this specification defines changes to an ISO standard, the ISO typographical conventions are used in Chapter 2 and Appendix A. Refer to Section 2.1 on page 3 for details.

Trade Marks

UNIX[®] is a registered trade mark of UNIX System Laboratories, Inc. in the U.S.A. and other countries.

X/Open[™] and the “X” device are trade marks of X/Open Company Limited in the U.K. and other countries.

/ Acknowledgements

X/Open gratefully acknowledges the collaboration with and contributions from the SQL Access Group.

Referenced Documents

The following standards are referenced in this specification:

ISO/IEC RDA Generic

ISO/IEC 9579-1: 1993, Information Technology — Open Systems Interconnection — Remote Database Access — Part 1: Generic Model, Service, and Protocol.

ISO/IEC RDA SQL

ISO/IEC 9579-2: 1993, Information Technology — Open Systems Interconnection — Remote Database Access — Part 2: SQL Specialization.

ISO 8823: 1988

ISO 8823: 1988, Information Processing Systems — Open Systems Interconnection — Connection Oriented Presentation Protocol Specification.

ISO 8859-1: 1987

ISO 8859-1: 1987, Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.

ISO/IEC SQL-92

ISO/IEC 9075: 1992, Information Technology — Database Language SQL.

OIW RDA

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 19 — Remote Database Access, Output from the December 1992 OSE Implementors' Workshop.

The following X/Open document is referenced in this specification:

X/Open SQL

X/Open CAE Specification, August 1992, Structured Query Language (SQL) (ISBN: 1-872630-58-8 C201).

The following documents are useful as supplementary reading but are not directly referenced:

ISO 8326: 1987

ISO 8326: 1987, Information Processing Systems — Open Systems Interconnection — Basic Connection Oriented Session Service Definition.

ISO 8649: 1988

ISO 8649: 1988, Information Processing Systems — Open Systems Interconnection — Service Definition for the Association Control Service Element.

ISO 8822: 1988

ISO 8822: 1988, Information Processing Systems — Open Systems Interconnection — Connection Oriented Presentation Service Definition.

ISO 8824: 1990

ISO 8824: 1990, Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).

Introduction

X/Open SQL Remote Database Access (RDA) provides access to one or more databases on a remote system. This specification discusses implementation issues in communication between an X/Open-compliant RDA client and RDA server.

The ISO/IEC RDA Generic standard defines a generic model, service and protocol for Remote Database Access (RDA). The ISO/IEC RDA SQL standard defines a specialisation to allow use with SQL. X/Open endorses these standards and incorporates them by reference. Refer to these standards for definitions of terms and for additional supporting material not repeated here. In this document a reference to “RDA standards” means both the ISO/IEC RDA Generic standard and the ISO/IEC RDA SQL standard.

Various standardisation and implementors’ groups have acknowledged that additional specifications or implementation agreements are needed to ensure interoperability. They are summarised in this document. X/Open expects that ISO and other standardisation groups will incorporate this material, or comparable material addressing these issues, into formal standards and Implementation Agreements.

Changes to the Referenced Documents

This chapter specifies changes to the ISO/IEC RDA Generic standard and the ISO/IEC RDA SQL standard that are required to support the features of ISO/IEC SQL-92 beyond Entry Level included in the X/Open **SQL** specification.

Applying the changes in this chapter to the referenced ISO documents produces a conceptual document called **X/Open RDA**.

Specific items in this section will be removed from future editions if they are eventually adopted by ISO for the RDA SQL-92 Addendum.

Some changes specified in this chapter affect Table 10 of the ISO/IEC RDA SQL standard. Appendix A of this document contains a revised Table 10 based on all these proposals.

Some changes specified in this chapter affect the ASN.1 module for RDA. Appendix B of this document contains a revised ASN.1 module based on all these proposals.

2.1 Typographical Equivalence

X/Open RDA implicitly refers to the X/Open **SQL** specification to define the SQL statement syntax allowed in an **R-ExecuteDBL** request. Table 10 in Clause 4.1.7.1.1 of the ISO/IEC RDA SQL standard uses a different typography from the X/Open **SQL** specification to refer to syntactic elements of SQL. This chapter follows the ISO typography, with some additions. That is:

- A syntactic element in angle brackets in the ISO/IEC RDA SQL standard is the same as the corresponding syntactic element in italics in the X/Open **SQL** specification. The use of punctuation in the respective documents is not significant. For example, <update statement: positioned> in the ISO/IEC RDA SQL standard is the same as *update-statement-positioned* in the X/Open **SQL** specification.
- The syntactic elements <table definition> and <view definition> in the ISO/IEC RDA SQL standard correspond respectively to *create-table-statement* and *create-view-statement* in the X/Open **SQL** specification.
- The syntactic elements <create index statement> and <drop index statement> are introduced by this chapter and correspond respectively to *create-index-statement* and *drop-index-statement* in the X/Open **SQL** specification.

2.2 sQLDBLStatementNotAllowed Error

In the current ISO/IEC RDA SQL standard, only transaction management statements cause an error if they are transmitted to the server. To allow for additional statements that also cause this error, apply the following changes to ISO/IEC RDA SQL standard:

1. Replace all occurrences of “sQLDBLTransactionStatementNotAllowed” with “sQLDBLStatementNotAllowed”.
2. Replace all occurrences of “SQLDBLTransactionStatementNotAllowed” with “SQLDBLStatementNotAllowed”.
3. In Clause 3.1.5.1.1, R-ExecuteDBLService, under Error Parameters, replace the description of sQLDBLTransactionStatementNotAllowed with the following:

The content of the sQLDBLStatement is one of the following statements not permitted by the RDA SQL Specialization:

- <allocate descriptor statement>
- <commit statement>
- <connect statement>
- <deallocate descriptor statement>
- <disconnect statement>
- <get descriptor statement>
- <get diagnostics statement>
- <rollback statement>
- <set connection statement>
- <set descriptor statement>

4. In Clause 4.1.7.1.1:

Delete the second sentence of Note 1 in Table 10 (R-ExecuteDBL use of SQL argument and result parameters).

Under Error Rules, replace the Predicate for sQLDBLTransactionStatementNotAllowed with:

The RDA client requested the execution of an RDA SQL statement that is one of the SQL statements listed in Table 10 (R-ExecuteDBL use of SQL argument and result parameters) with Note 1.

2.3 SQL Diagnostics Information

Apply the following changes to the ISO/IEC RDA SQL standard:

1. In Clause 1.3.1 (terms defined in ISO SQL), add:

<get diagnostics statement>

2. In Clause 3.1.4.1.1 (R-Open):

Add the following as a third subparameter of sQLOpenArgument in the R-Open SQL Specific Service Parameters table:

	Req	Ind	Rsp	Cnf
sQLDiagnosticsRequested	U	C(=)		

Add the following description after sQLConformanceLevel in the Request Parameters section:

sQLDiagnosticsRequested:

This parameter specifies the level of diagnostics information requested by the RDA client. If “always” is chosen, then the RDA client desires that sQLDiagnostics always be included in the results of subsequent R-ExecuteDBL requests. If “onRequest” is chosen, then the RDA client desires that sQLDiagnostics be included in the results of a subsequent R-ExecuteDBL request only when specifically requested on that individual R-ExecuteDBL request. If “never” is chosen, then the RDA client does not desire that sQLDiagnostics be included in the results of subsequent R-ExecuteDBL requests. The default value for sQLDiagnosticsRequested is “never”.

3. In Clause 3.1.5.1.1 (R-ExecuteDBL):

Add the following request parameter to the end of the list of request parameters in the R-ExecuteDBL SQL Specific Service Parameters table:

	Req	Ind	Rsp	Cnf
returnSQLDiagnostics	U	C(=)		

Add the following description after listOfSQLDBLArgumentValues in the Request Parameters section:

returnSQLDiagnostics:

This parameter specifies the level of diagnostics information requested by the RDA client when “onRequest” was chosen by the RDA client for the sQLDiagnosticsRequested parameter on R-Open. If “true” is chosen, then the RDA client desires that sQLDiagnostics be included in the results of the request. If “false” is chosen, then the RDA client desires that sQLDiagnostics not be included in the results of the request. The default value for returnSQLDiagnostics is “false”.

Add the following subparameter after the sQLErrorText subparameter of the sQLDBLException result parameter in the R-ExecuteDBL SQL Specific Service Parameters table:

	Req	Ind	Rsp	Cnf
sQLDiagnostics			U	C(=)

Add the following description after `SQLExceptionText` in the Result Parameters section:

sQLDiagnostics:

The semantics of the components of `sQLDiagnostics` are specified in Clause 18 (Diagnostics Management) of ISO/IEC 9075:1992. An optional parameter may be omitted if its value is null.

4. Add the following attribute to Clause 4.1.1.2 (Opened Data Resource Entity):

sQLDiagnosticsRequested:

This attribute specifies the level of diagnostics information requested by the RDA client.

“always”:

The RDA client desires that `sQLDiagnostics` always be included in the results of subsequent R-ExecuteDBL requests.

“onRequest”:

The RDA client desires that `sQLDiagnostics` be included in the results of a subsequent R-ExecuteDBL request only when specifically requested on that individual R-ExecuteDBL request.

“never”:

The RDA client does not desire that `sQLDiagnostics` be included in the results of subsequent R-ExecuteDBL request.

5. Add the following to the entity manipulation rules in Clause 4.1.6.1.1 (R-Open Service):

Attribute	Initial Value
<code>sQLDiagnosticsRequested</code>	The <code>sQLDiagnosticsRequested</code> parameter value on the R-Open indication primitive, if provided. Otherwise, this attribute contains the default value “never”.

6. Add the following to the result rule for `sQLDBLException` in Clause 4.1.7.1.1 (R-ExecuteDBL Service):

If the `sQLDiagnosticsRequested` attribute of the opened data resource entity has a value of “always”, then `sQLDiagnostics` shall be returned by the RDA server. If the `sQLDiagnosticsRequested` attribute of the opened data resource entity has a value of “onRequest” and the `returnSQLDiagnostics` parameter of the R-ExecuteDBL indication has a value of “true”, then `sQLDiagnostics` shall be returned by the RDA server. Otherwise, `sQLDiagnostics` shall not be returned.

7. Add the following row to Table 10 (R-ExecuteDBL use of SQL argument and result parameters):

RDA SQL Statement To Be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<code><get diagnostics statement>¹</code>				

8. In Clause 4.2.2 (ASN.1 Module):

Replace SQLOpenArgument with the following:

```
SQLOpenArgument ::= SEQUENCE
{
  charSet [0] OBJECT IDENTIFIER OPTIONAL,
  sqlConformanceLevel [1] OBJECT IDENTIFIER OPTIONAL,
  sqlDiagnosticsRequested [2] ENUMERATED
  {
    always (0),
    onRequest (1),
    never (2)
  } DEFAULT never
}
```

Add the following parameter at the end of the SEQUENCE for R-ExecuteDBL-Request:

```
returnSQLDiagnostics [6] BOOLEAN DEFAULT TRUE
```

(Add a comma after the OPTIONAL that concludes dblArguments.)

Replace SQLDBLException with the following:

```
SQLDBLException ::= SEQUENCE
{
  sqlSTATE [0] VisibleString OPTIONAL,
  sqlCODE [1] INTEGER OPTIONAL,
  sqlErrorText [2] VisibleString OPTIONAL,
  sqlDiagnostics [3] SQLDiagnostics OPTIONAL
}
```

```
SQLDiagnostics ::= SEQUENCE
{
  rowCount [0] INTEGER OPTIONAL,
  exceptionList [3] SEQUENCE OF ExceptionInfo
}
```

```
ExceptionInfo ::= SEQUENCE
{
  returnedSQLSTATE [0] VisibleString,
  classOrigin [1] VisibleString,
  subclassOrigin [2] VisibleString,
  messageText [3] VisibleString OPTIONAL
}
```

2.4 VARCHAR Data Type

Apply the following changes to the ISO/IEC RDA SQL standard:

1. Append the following entry to Table 8 (sQLDBLArgumentSpecification and sQLDBLResultSpecification):

	Req	Ind	Rsp	Cnf
varcharType	S	S(=)	S	S(=)
charSet	U	C(=)	U	C(=)
length	M	M(=)	M	M(=)

2. Append the following to Clause 3.1.6.2 (sQLDBLArgumentSpecification and sQLDBLResultSpecification):

varcharType:

This parameter describes an item of varying length character data.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for character data associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. If the parameter is omitted, then the character set is the default established by the declaration during the execution of the R-Open service that opened the associated SQL database resource. If no default was established, then this parameter must be specified.

length:

This parameter specifies the maximum number of characters allowed for the corresponding varying length character data item.

3. Add the following entry after doublePrecisionItem in Table 9 (sQLDBLArgumentValues and sQLDBLResultValues):

	Req	Ind	Rsp	Cnf
varcharItem	S	S(=)	S	S(=)

4. Add the following after doublePrecisionItem description in Clause 3.1.6.3 (sQLDBLArgumentValues and sQLDBLResultValues):

varcharItem:

This parameter contains the value of a varying length character Data Variable. The encoding of the character data shall be the encoding specified in the coded character set specification identified by the corresponding charSet parameter.

5. In Clause 4.2.2 (ASN.1 Module), extend the typeDescriptor subparameter of the SQLDataTypeDescriptor parameter as follows:

```

varcharType                [15] SEQUENCE
-- SQL type: varchar
{ charSet                   OBJECT IDENTIFIER OPTIONAL,
  length                     INTEGER
}
```

6. In Clause 4.2.2 (ASN.1 Module), extend the dataItem subparameter of the SQLValue parameter as follows:

varcharItem

[10] OCTET STRING

2.5 Dynamic SQL Statements

The ISO/IEC RDA SQL standard does not include support for the dynamic SQL statements provided by the X/Open SQL specification. To include support for these statements, apply the changes below to Table 10 (R-ExecutedDBL use of SQL argument and result parameters) of the ISO/IEC RDA SQL standard.

Add the following rows:

RDA SQL Statement To Be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<allocate descriptor statement> ¹ <deallocate descriptor statement> ¹ <describe statement> <dynamic close statement> <dynamic declare cursor> <dynamic delete statement: positioned> <dynamic fetch statement> <dynamic open statement> <dynamic update statement: positioned> <execute statement> <execute immediate statement> <get descriptor statement> ¹ <prepare statement> <set descriptor statement> ¹	C ¹¹		C ← S ⁷ C → S ^{8,10} C ← S ⁹	C ← S

In X/Open RDA, SQL descriptor statements are processed locally at the RDA client. The RDA client does not send to the server any of the four SQL statements: <allocate descriptor statement>, <deallocate descriptor statement>, <get descriptor statement> or <set descriptor statement>.

If a <dynamic fetch statement>, <dynamic open statement> or <execute statement> refers to an SQL descriptor area, the RDA client sends the SQLDataTypeDescriptors corresponding to the descriptor items of the referenced descriptor area. If a <describe statement> refers to the name of an SQL descriptor using an <embedded variable name>, the RDA client does not send the contents of that host variable.

The notes for the dynamic SQL statements added to Table 10 of the ISO/IEC RDA SQL standard, as shown in Appendix A, specify this behaviour.

2.6 Connection Management Statements

The ISO/IEC RDA SQL standard does not include support for the SQL connection management statements provided by the X/Open **SQL** specification. To include support for these statements, modify the ISO/IEC RDA SQL standard as follows:

1. Add the following rows to Table 10 (R-ExecuteDBL use of SQL argument and result parameters):

RDA SQL Statement To Be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<connect statement> ¹				
<disconnect statement> ¹				
<set connection statement> ¹				

2. In Clause 3.1.5.1.1, in the description of the sQLDBLStatementNotAllowed error (as modified by Section 2.2), insert the following statements in alphabetic order within the list:

<connect statement>
 <disconnect statement>
 <set connection statement>

2.7 Data Definition Statements

The ISO/IEC RDA SQL standard does not include support for all the SQL data definition statements provided by the X/Open **SQL** specification. There is also one SQL data definition statement in the ISO/IEC RDA SQL standard that is not provided by the X/Open **SQL** specification. To accommodate these differences, apply the following changes to Table 10 (R-ExecuteDBL use of SQL argument and result parameters) of the ISO/IEC RDA SQL standard:

1. Delete the following row:

<schema definition>

2. Add the following rows:

RDA SQL Statement To Be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<alter table statement>				
<create index statement>				
<drop index statement>				
<drop table statement>				
<drop view statement>				
<revoke statement>				

Implementation Agreements

This chapter describes implementation agreements at the protocol level on maximum values and on the use of parameters required with X/Open SQL.

3.1 Character Set

Clients and servers must support at least the characters from columns 2 to 7 inclusive of the character code chart in the ISO 8859-1 standard. To specify the character set of the ISO 8859-1 standard (in the `charSet` parameter of the R-Open service), clients and servers use the object identifier specified in Section 3.8.2 on page 33.

3.2 Operation Limits

An X/Open-compliant server can process a minimum of 32 pending (outstanding) RDA operations on a single RDA dialogue. It may reject an RDA operation for an RDA dialogue if it already has 32 operations pending on that dialogue.

3.3 Transactions

The number of data definition statements that a server can execute within a transaction is implementation-defined but is at least one. For maximum interoperability, clients should send only one data definition statement within a transaction.

3.4 RDA Features Not Used in This Specification

This specification does not use the following features of RDA:

- the RDA operations **R-DefineDBL**, **R-InvokeDBL** and **R-DropDBL**
- control services on another dialogue
- RDA SQL TP Application Context.

Future X/Open SQL RDA Specifications may make use of some or all of these features of RDA.

3.5 Protocol Stack

3.5.1 Session Layer

The session layer protocol requirements for RDA are as follows:

Session Version	2
Session Requirements	Duplex functional unit only
Calling Session Selector	4 octets
Called Session Selector	4 octets

3.5.2 Presentation Layer

The presentation layer protocol requirements for RDA are as follows:

Presentation Version	1
Presentation Requirements	Kernel unit only
Mode	1 (normal mode only)
Calling Presentation Selector	4 octets
Called Presentation Selector	4 octets

A Presentation Service user always selects a value of Fully-encoded-data in *User-data*, in conjunction with type **Single-ASN1-type** (see Clause 8.4.2 of ISO 8823:1988). It sets up an explicit Presentation Context Definition List using P-CONNECT, containing the two contexts listed below, and does not require default contexts.

Context	Abstract Syntax Object Identifier	Transfer Syntax Object Identifier
ACSE	{2 2 1 0 1}	{2 1 1}
RDA	See Section 3.8.1 on page 33.	{2 1 1}

A server informs a client, in implementation-defined ways, of the entire Presentation Address the client uses to establish a connection.

An X/Open-compliant implementation (client or server) can process RDA APDUs of up to 30,000 octets. However, X/Open recommends that implementations do not restrict the size of Presentation user data.

3.5.3 Association Control Service Element (ACSE)

The ACSE protocol requirements for RDA are as follows:

ACSE Version	1
Application Context Object Identifier	See Section 3.8 on page 33.

3.6 RDA Parameters

This section describes how X/Open SQL RDA clients and servers use RDA parameters. Tables describe each parameter's usage and any limits that apply. The tables are presented in the same order as in the RDA standards.

Limitations on Parameters

The **Parameter** column specifies the parameter by name. The dagger (†) symbol in the **Parameter** column indicates that the parameter is an X/Open extension to the ISO/IEC RDA SQL standard.

The **Limitation** column describes any limitation of the RI/RC APDU parameter value in addition to the limits imposed by the RDA standards. Where the limitation includes a maximum value, all X/Open-compliant servers allow at least that maximum. It is implementation-defined whether the server supports a larger maximum value. Ranges defined by text such as “from 1 to 32” are always inclusive. To ensure interoperability, clients should restrict parameter values to the range shown in the tables. If a parameter value is outside this range, X/Open does not specify the resulting behaviour. The limit of the length of an RDA APDU applies even when the limits for parameters within the same RDA APDU add up to a greater value.

The * symbol in the **Limitations** column indicates that the source of a limitation is the **Limits** section in the X/Open SQL specification. This section describes limitations of SQL implementations to which an application should adhere for portability.

Subparameters

Each RDA parameter appears on a separate line. Some lines describe subparameters (structures of component parameters). The structure is shown by the bullet (•) symbol in the **Parameter** column. A parameter name preceded by one or more bullets is a subparameter of the nearest entry above it that has one fewer bullet.

Mandatory Parameters

The ASN.1 module in Appendix B specifies whether a parameter is mandatory or optional. For those parameters that the RDA standards also define, in no case does the X/Open SQL specification differ from the RDA standards as to whether a parameter is mandatory or optional.

Unused Parameters

When the limitations below specify that a server **ignores** a parameter, it means that X/Open does not define the parameter's usage. The server must not reject a request based on the presence, absence or contents of the parameter.

Magnitude of INTEGER Parameters

Unless otherwise specified in the following tables, an INTEGER parameter is limited to a value from -2,147,483,648 to 2,147,483,647 (maximum 4 octets).

3.6.1 R-Initialize Service

Table 3-1 Parameters for R-Initialize request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
dialogueID	See subparameters below.
•dialogueIDClientInvocation	See subparameters below.
••aP-title	No additional limitation.
••aE-qualifier	No additional limitation.
••aP-invocationID	No additional limitation.
••aE-invocationID	No additional limitation.
•dialogueIDSuffix	An OCTET STRING from 1 to 64 octets long.
identityOfUser	A VisibleString from 1 to 64 characters long.
userAuthenticationData	The client may provide an IA5String from 1 to 255 characters long, an OCTET STRING from 1 to 255 octets long, or a BIT STRING from 1 to 2040 bits long. If the server does not use this parameter, it ignores the parameter.
controlServiceDataRequested	A BOOLEAN whose value should be FALSE since X/Open-compliant servers are not required to support control services on another dialogue.
functionalUnitsRequested	The client should not request the Stored Execution DBL functional unit. X/Open-compliant implementations are not required to support the Stored Execution DBL functional unit.
SQLInitializeArgument	See subparameters below.
•SQLConformanceLevelDefault	An OBJECT IDENTIFIER. See Section 3.8 on page 33.
•userData	An OCTET STRING from 1 to 255 octets long. X/Open does not specify its content.

Table 3-2 Parameters for R-Initialize result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
controlServiceData	See subparameters below.
•controlServicesAllowed	No additional limitation.
•controlAuthenticationData	The server may provide an IA5String from 1 to 255 characters long, an OCTET STRING from 1 to 255 octets long, or a BIT STRING from 1 to 2040 bits long.
functionalUnitsAllowed	No additional limitation.
sqlInitializeResult	See subparameters below.
•userData	An OCTET STRING from 1 to 255 octets long. X/Open does not specify its content.

Table 3-3 Parameters for R-Initialize error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
accessControlViolation	No additional limitation.
duplicateDialogueID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
userAuthenticationFailure	No additional limitation.

3.6.2 R-Synchronize APDU

The R-Synchronize-RI APDU has no parameters.

3.6.3 R-Terminate Service**Table 3-4** Parameters for R-Terminate request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-5 Parameters for R-Terminate result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-6 Parameters for R-Terminate error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
serviceNotNegotiated	No additional limitation.

3.6.4 R-BeginTransaction Service

Table 3-7 Parameters for R-BeginTransaction request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-8 Parameters for R-BeginTransaction error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
serviceNotNegotiated	No additional limitation.

3.6.5 R-Commit Service

Table 3-9 Parameters for R-Commit request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-10 Parameters for R-Commit result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
transactionResult	No additional limitation.

Table 3-11 Parameters for R-Commit error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.

3.6.6 R-Rollback Service

Table 3-12 Parameters for R-Rollback request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-13 Parameters for R-Rollback result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-14 Parameters for R-Rollback error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.

3.6.7 R-Cancel Service

Table 3-15 Parameters for R-Cancel request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
controlledDialogue	This parameter and its subparameters should be omitted because X/Open-compliant servers are not required to support controlled dialogues.
•dialogueID	See controlledDialogue.
••dialogueIDClientInvocation	See controlledDialogue.
•••aP-title	See controlledDialogue.
•••aE-qualifier	See controlledDialogue.
•••aP-invocationID	See controlledDialogue.
•••aE-invocationID	See controlledDialogue.
••dialogueIDSuffix	See controlledDialogue.
•controlAuthenticationData	See controlledDialogue.
listOfOperationID	This list may contain from 1 to 32 elements.
•OperationID	An INTEGER with value greater than 0.

Table 3-16 Parameters for R-Cancel result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.

Table 3-17 Parameters for R-Cancel error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
controlAuthenticationFailure	No additional limitation.
controlServicesNotAllowed	No additional limitation.
dialogueIDUnknown	No additional limitation.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
serviceNotNegotiated	No additional limitation.

3.6.8 R-Status Service

Table 3-18 Parameters for R-Status request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
controlledDialogue	This parameter and its subparameters should be omitted because X/Open-compliant servers are not required to support controlled dialogues.
•dialogueID	See controlledDialogue.
••dialogueIDClientInvocation	See controlledDialogue.
•••aP-title	See controlledDialogue.
•••aE-qualifier	See controlledDialogue.
•••aP-invocationID	See controlledDialogue.
•••aE-invocationID	See controlledDialogue.
••dialogueIDSuffix	See controlledDialogue.
•controlAuthenticationData	See controlledDialogue.
listOfOperationID	This list may contain from 1 to 32 elements.
•OperationID	An INTEGER with value greater than 0.

Table 3-19 Parameters for R-Status result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
listOfStatusInformation	This list may contain from 1 to 32 elements.
•StatusInformation	See subparameters below.
••operationID	An INTEGER with value greater than 0.
••operationStatus	See choices below.
•••operationIDUnknown	No additional limitation.
•••awaitingExecution	No additional limitation.
•••executing	No additional limitation.
•••finished	No additional limitation.
•••cancelled	No additional limitation.
•••aborted	A VisibleString from 1 to 254 characters long.

Table 3-20 Parameters for R-Status error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
controlAuthenticationFailure	No additional limitation.
controlServicesNotAllowed	No additional limitation.
dialogueIDUnknown	No additional limitation.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
serviceNotNegotiated	No additional limitation.

3.6.9 R-Open Service

Table 3-21 Parameters for R-Open request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
dataResourceHandle	An INTEGER with value greater than 0.
dataResourceName	A VisibleString from 1 to 254 characters long.
sQLAccessControlData	The client may provide an IA5String from 1 to 255 characters long, an OCTET STRING from 1 to 255 octets long, or a BIT STRING from 1 to 2040 bits long. If the server does not use this parameter, it ignores the parameter.
sQLUsageMode	No additional limitation.
sQLOpenArgument	See subparameters below.
•charSet	An OBJECT IDENTIFIER from 2 to 16 elements long.
•sQLConformanceLevel	An OBJECT IDENTIFIER. See Section 3.8 on page 33.
•sQLDiagnosticsRequested †	No additional limitation.

Table 3-22 Parameters for R-Open result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
sQLOpenResult	See subparameters below.
•charSet	An OBJECT IDENTIFIER from 2 to 16 elements long.
•charSetNotSupported	No additional limitation.
•sQLConformanceLevel	An OBJECT IDENTIFIER. See Section 3.8 on page 33.

Table 3-23 Parameters for R-Open error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
dataResourceAlreadyOpen	See subparameter below.
•dataResourceHandle	An INTEGER with value greater than 0.
dataResourceNameNotSpecified	No additional limitation.
dataResourceNotAvailable	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
dataResourceUnknown	No additional limitation.
duplicateDataResourceHandle	No additional limitation.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
operationCancelled	No additional limitation.
serviceNotNegotiated	No additional limitation.
SQLOpenError	See choices below.
•invalidSQLConformanceLevel	No additional limitation.
•rDATransactionOpen	No additional limitation.
•SQLAccessControlViolation	No additional limitation.
•SQLDatabaseResourceAlreadyOpen	No additional limitation.

3.6.10 R-Close Service

Table 3-24 Parameters for R-Close request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
listOfDataResourceHandle	This list must contain only one element.
•DataResourceHandle	An INTEGER with value greater than 0.

Table 3-25 Parameters for R-Close result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
listOfCloseExceptions	This list must contain only one element.
•CloseException	See subparameters below.
••dataResourceHandle	An INTEGER with value greater than 0.
••closeException	See choice below.
•••dataResourceHandleUnknown	No additional limitation.

Table 3-26 Parameters for R-Close error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
operationCancelled	No additional limitation.
serviceNotNegotiated	No additional limitation.
sqlCloseError	See choice below.
•rDATransactionOpen	No additional limitation.

3.6.11 R-ExecuteDBL Service

Table 3-27 Parameters for R-ExecuteDBL request

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
dataResourceHandle	An INTEGER with value greater than 0.
SQLDBLStatement	See subparameters below.
•statementText	An OCTET STRING from 1 to 4000* octets long.
•charSet	An OBJECT IDENTIFIER from 2 to 16 elements long.
SQLDBLArgumentSpecification	This parameter may contain from 1‡ to 100 entries of SQLDataTypeDescriptor. See Table 3-30 on page 30.
SQLDBLResultSpecification	This parameter may contain from 1‡ to 100 entries of SQLDataTypeDescriptor. See Table 3-30 on page 30.
dBLArguments	See choices below.
•singleArgument	See subparameters below.
••repetitionCount	An INTEGER with value from 1 to 64.
••SQLDBLArgumentValues	This parameter may contain from 1‡ to 100 entries of SQLValue. See Table 3-32 on page 32.
•multipleArguments	See subparameters below.
••listOfSQLDBLArgumentValues	This list may contain from 1 to 64 elements.
•••SQLDBLArgumentValues	This parameter may contain from 1‡ to 100 entries of SQLValue. See Table 3-32 on page 32.
returnsQLDiagnostics †	No additional limitation.

Table 3-28 Parameters for R-ExecuteDBL result response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
SQLDBLResultSpecification	This parameter may contain from 1‡ to 100 entries of SQLDataTypeDescriptor. See Table 3-30 on page 30.
listOfResultValues	This list may contain from 1 to 64 elements.
•ResultValues	See subparameters below.
••SQLDBLException	See Table 3-31 on page 31.
•••SQLDBLResultValues	This parameter may contain from 1‡ to 100 entries of SQLValue. See Table 3-32 on page 32.

‡ For maximum interoperability, the sender should not send this parameter with zero elements. However, the receiver should accept this parameter with zero elements and process it as though the sender omitted the parameter.

Table 3-29 Parameters for R-ExecuteDBL error response

Parameter	Limitation
operationID	An INTEGER with value greater than 0.
badRepetitionCount	No additional limitation.
dataResourceHandleNotSpecified	No additional limitation.
dataResourceHandleUnknown	No additional limitation.
duplicateOperationID	No additional limitation.
invalidSequence	See subparameter below.
•diagnosticInformation	No additional limitation.
noDataResourceAvailable	No additional limitation.
operationAborted	See subparameters below.
•errorType	No additional limitation.
•diagnosticInformation	A VisibleString from 1 to 254 characters long.
operationCancelled	No additional limitation.
serviceNotNegotiated	No additional limitation.
transactionRolledBack	No additional limitation.
sQLExecuteDBLError	See choices below.
•hostIdentifierError	No additional limitation.
•rDATransactionNotOpen	No additional limitation.
•sQLDBLArgumentCountMismatch	No additional limitation.
•sQLDBLArgumentTypeMismatch	No additional limitation.
•sQLDBLNoCharSet	No additional limitation.
•sQLDBLStatementNotAllowed †	No additional limitation.
•sQLUsageModeViolation	No additional limitation.

3.7 Limits for Common Parameters

This section describes the parameters of common RDA types.

Table 3-30 Parameters for SQLDataTypeDescriptor

Parameter	Limitation
nullable	No additional limitation.
colName	A VisibleString from 1 to 18* characters long.
typeDescriptor	See choices below.
•characterType	See subparameters below.
••charSet	An OBJECT IDENTIFIER from 2 to 16 elements long.
••length	An INTEGER representing the maximum number of total text characters in the data value. Trailing blanks are included in the length, but any trailing null byte is not. If this SQLDataTypeDescriptor defines the data type of an argument value that contains an SQLDBLStatement, the value does not exceed 4000*. Otherwise, the value is from 1 to 254*.
••fixedLengthEncoding	No additional limitation.
•numericType	See subparameters below.
••precision	This value must be from 1 to 15.
••scale	This value must be from 0 to the value of precision.
•decimalType	See subparameters below.
••precision	This value must be from 1 to 15.
••scale	This value must be from 0 to the value of precision.
•integerType	See subparameters below.
••precision	If precisionBase indicates that precision is a number of decimal digits, then precision must be in the range from 1 to 10*. If precisionBase indicates that precision is a number of binary digits, then precision must be in the range from 1 to 31*.
••precisionBase	No additional limitation.
•smallIntType	See subparameters below.
••precision	If precisionBase indicates that precision is a number of decimal digits, then precision must be in the range from 1 to 5*. If precisionBase indicates that precision is a number of binary digits, then precision must be in the range from 1 to 15*.
••precisionBase	No additional limitation.
•floatType	See subparameters below.
••mantissaPrecision	An INTEGER with value from 1 to 47*.
••maxExponent	An INTEGER with value from 0 to 38*.
•realType	See subparameters below.
••mantissaPrecision	An INTEGER with value from 1 to 21*.
••maxExponent	An INTEGER with value from 0 to 38*.
•doublePrecisionType	See subparameters below.
••mantissaPrecision	An INTEGER with value from 1 to 47*.

Parameter	Limitation
<ul style="list-style-type: none"> ••maxExponent •varCharType † ••charSet † ••length † 	<p>An INTEGER with value from 0 to 38*.</p> <p>See subparameters below.</p> <p>An OBJECT IDENTIFIER from 2 to 16 elements long.</p> <p>An INTEGER representing the maximum number of total text characters in the data value. Trailing blanks are included in the length, but any trailing null byte is not. If this <code>SQLDataTypeDescriptor</code> defines the data type of an argument value that contains an <code>SQLDBLStatement</code>, the value does not exceed 4000*. Otherwise, the value is from 1 to 254*.</p>

Table 3-31 Parameters for SQLDBLException

Parameter	Limitation
<code>SQLState</code>	A server returns this 5-character <code>VisibleString</code> on any <code>SQLDBLException</code> it returns from R-ExecuteDBL . For valid values, see the Appendix B of the X/Open SQL specification.
<code>SQLCode</code>	This parameter corresponds to the deprecated <code>SQLCODE</code> status variable. Servers may return <code>SQLCode</code> in addition to <code>SQLState</code> .
<code>SQLErrorText</code>	A server may return a <code>VisibleString</code> from 1 to 254* characters long on any <code>SQLDBLException</code> it returns from R-ExecuteDBL .
<code>SQLDiagnostics</code> †	See subparameters below.
•rowCount †	No additional limitation.
•exceptionList †	This list may contain from 1 to 10 entries of <code>exceptionInfo</code> .
••exceptionInfo †	See subparameters below.
•••returnedSQLState †	A 5-character <code>VisibleString</code> . For valid values, see Appendix B of the X/Open SQL specification.
•••classOrigin †	A <code>VisibleString</code> from 1 to 254* characters long. Valid values correspond to those for the <code>CLASS_ORIGIN</code> field of the diagnostics area. See Section 5.8.1 of the X/Open SQL specification.
•••subclassOrigin †	A <code>VisibleString</code> from 1 to 254* characters long. Valid values correspond to those for the <code>SUBCLASS_ORIGIN</code> field of the diagnostics area. See Section 5.8.1 of the X/Open SQL specification.
•••messageText †	A <code>VisibleString</code> from 1 to 254* characters long.

Table 3-32 Parameters for SQLValue

Parameter	Limitation
dataItem	See choices below.
•characterItem	An OCTET STRING from 1 to 254* characters long.
•numericItem	An INTEGER with absolute value less than 10 ¹⁵ *.
•decimalItem	An INTEGER with absolute value less than 10 ¹⁵ *.
•integerItem	An INTEGER with a value from -2,147,483,648* to 2,147,483,647*.
•smallIntItem	An INTEGER with a value from -32,768* to 32,767*.
•floatItem	A REAL with a value of 0 or absolute value within the range of 10 ⁻³⁸ * to 10 ³⁸ *.
•realItem	A REAL with a value of 0 or absolute value within the range of 10 ⁻³⁸ * to 10 ³⁸ *.
•doublePrecisionItem	A REAL with a value of 0 or absolute value within the range of 10 ⁻³⁸ * to 10 ³⁸ *.
•varcharItem †	An OCTET STRING from 0 to 254* characters long.
indicator	No additional limitation.

3.8 Object Identifiers

This specification defines certain object identifiers for cases in which it is necessary to distinguish between the ISO/IEC RDA SQL standard and X/Open RDA. It also references object identifiers defined by other authorities.

3.8.1 Object Identifiers Defined By X/Open

All X/Open-defined object identifiers begin with the following elements:

```
{ iso (1) national-member-body (2) uk (826)
  national (0) x-open (1050) data-management (3) ...
```

The following object identifiers are used in this specification:

Usage	Object Identifier
sQLConformanceLevel (without optional integrity feature)	... x-open-sql (1) year-1992 (1) full-conformance (3) integrity-no (0) }
sQLConformanceLevel (with optional integrity feature)	... x-open-sql (1) year-1992 (1) full-conformance (3) integrity-yes (1) }
Application Context Name*	... x-open-rdasql (2) basic-ac (2) version-1 (1) }
RDA Abstract Syntax*	... x-open-rdasql (2) abstract-syntax (1) version-1 (1) }
ASN.1 Module*	... x-open-rdasql (2) module (0) version-1 (1) }

An X/Open-compliant implementation must support the X/Open-defined object identifiers for Application Context Name, RDA Abstract Syntax and ASN.1 Module. In addition, an X/Open-compliant implementation must indicate its level of functionality through the use of the sQLConformanceLevel object identifier.

3.8.2 Object Identifiers Defined By Other Authorities

The following object identifier defined by another authority is also used in this specification:

Usage	Object Identifier
charSet **	{ iso (1) identified-organization (3) oiw (14) rda-sig (9) character-sets (1) oiw-latin-1 (1) abstract-syntax (1) }

The meaning of this object identifier is defined in OIW RDA.

* When the features required to support X/Open SQL RDA are provided by the RDA standards, X/Open-compliant implementations will be expected to use the object identifiers from these standards.

** X/Open-compliant implementations use this object identifier to identify a character set. If ISO should select an object identifier to refer to this character set, X/Open-compliant implementations are expected to use it instead of the object identifier in this specification.

3.9 Prospective Uses of Parameters

The `repetitionCount` parameter in the **R-ExecuteDBL** request directs the server to repeat the execution of the SQL statement the specified number of times, using the same argument. A `repetitionCount` greater than 1 might be useful when executing:

- INSERT
- FETCH (both static and dynamic)
- EXECUTE (or EXECUTE IMMEDIATE).

(Currently, there is no occasion in the X/Open SQL application interface that would call for use of this feature.)

The `listOfSQLDBLArgumentValues` parameter in the **R-ExecuteDBL** request directs the server to repeat the execution of the SQL statement the specified number of times, using a successive argument from the list each time. This parameter might be useful when executing:

- INSERT
- Searched DELETE
- Searched UPDATE
- EXECUTE (if the statement to be executed is one of the above)
- SELECT.

(Currently, there is no occasion in the X/Open SQL application interface that would call for use of this feature.)

Resulting Text of Table 10

Parts of Chapter 2 of this document specify changes to Table 10 (R-ExecuteDBL use of SQL argument and result parameters) of the ISO/IEC RDA SQL standard. This appendix shows the resulting text of Table 10. Change bars in the right margin identify the X/Open changes.

Table 10. R-Execute use of SQL argument and result parameters.				
RDA SQL Statement to be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<allocate descriptor statement> ¹				
<alter table statement>				
<close statement>				
<commit statement> ¹				
<connect statement> ¹				
<create index statement>				
<deallocate descriptor statement> ¹				
<declare cursor>	$C \rightarrow S (H)^6$			
<delete statement: positioned>				
<delete statement: searched>	$C \rightarrow S (H)$	$C \rightarrow S (H)$		
<describe statement>	11		$C \leftarrow S^7$	
<disconnect statement> ¹				
<drop index statement>				
<drop table statement>				
<drop view statement>				
<dynamic close statement>				
<dynamic declare cursor>				
<dynamic delete statement: positioned>				
<dynamic fetch statement>			$C \rightarrow S^{8,10}$	$C \leftarrow S$
<dynamic open statement>	$C \rightarrow S (H)^{10}$	$C \rightarrow S (H)$	$C \leftarrow S^9$	
<dynamic update statement: positioned>	$C \rightarrow S (H)$	$C \rightarrow S (H)$		
<execute statement>	$C \rightarrow S (H)^{10}$	$C \rightarrow S (H)$		
<execute immediate statement>	$C \rightarrow S$	$C \rightarrow S$		
<fetch statement>			$C \rightarrow S^2$	$C \leftarrow S$
<get descriptor statement> ¹				
<get diagnostics statement> ¹				
<grant statement>				
<insert statement>	$C \rightarrow S (H)$	$C \rightarrow S (H)$		
<open statement>	$C \rightarrow S (H)^5$	$C \rightarrow S (H)$	$C \leftarrow S^3$	
<prepare statement>	$C \rightarrow S$	$C \rightarrow S$		
<rollback statement> ¹				
<revoke statement>				
<select statement: single row>	$C \rightarrow S (H)$	$C \rightarrow S (H)$	$C \leftrightarrow S^4$	$C \leftarrow S$
<set connection statement> ¹				
<set descriptor statement> ¹				
<table definition>				
<update statement: positioned>	$C \rightarrow S (H)$	$C \rightarrow S (H)$		
<update statement: searched>	$C \rightarrow S (H)$	$C \rightarrow S (H)$		
<view definition>				

Legend:	
ArgSpec	The <code>SQLDBLArgumentSpecification</code> parameter.
ArgVal	The <code>SQLDBLArgumentValues</code> parameter.
ResSpec	The <code>SQLDBLResultSpecification</code> parameter.
ResVal	The <code>SQLDBLResultValues</code> parameter.
C → S	Parameter supplied by the client.
C ← S	Parameter returned by the server.
(H)	The client must set this parameter if the RDA SQL statement contains host variables, except as noted below.
blank cells	unspecified (that is, not subject to conformance testing)

Notes:	
1	See the <code>SQLDBLStatementNotAllowed</code> error under the Error Rules below.
2	The client may optionally send the <code>ResSpec</code> , which will supersede the <code>ResSpec</code> sent on a previously executed (via an <code>R-ExecuteDBL</code> operation) <code><fetch statement></code> using the same cursor or received from the server on the corresponding <code><open statement></code> .
3	The <code>ResSpec</code> describes the <code>ResVal</code> that the subsequently executed (via an <code>R-ExecuteDBL</code> operation) <code><fetch statement></code> using this cursor will return, but any <code>ResSpec</code> specified with such a <code><fetch statement></code> overrides this <code>ResSpec</code> .
4	The client may optionally send the <code>ResSpec</code> , otherwise the server shall return the <code>ResSpec</code> which describes the <code>ResVal</code> , if any.
5	The client may optionally send the <code>ArgSpec</code> . If it is not specified, the server shall use the <code>ArgSpec</code> sent on the previously executed (via an <code>R-ExecuteDBL</code> operation) <code><declare cursor></code> statement that used the same cursor name.
6	The client may optionally send the <code>ArgSpec</code> . If it is not specified, the client shall send the <code>ArgSpec</code> on the subsequently executed (via an <code>R-ExecuteDBL</code> operation) <code><open statement></code> that uses the same cursor name.
7	The RDA server shall return a <code>ResSpec</code> only if the <code><SQL statement name></code> in the <code><describe statement></code> refers to a prepared statement that is a <code><cursor specification></code> . The <i>n</i> th <code>SQLDataTypeDescriptor</code> of the <code>ResSpec</code> contains the description of the <i>n</i> th column of the table defined by that prepared statement.
8	The client may optionally send the <code>ResSpec</code> , which will supersede the <code>ResSpec</code> sent on a previously executed (via an <code>R-ExecuteDBL</code> operation) <code><dynamic fetch statement></code> using the same cursor or received from the server on the corresponding <code><dynamic open statement></code> .
9	The <code>ResSpec</code> describes the <code>ResVal</code> that the subsequently executed (via an <code>R-ExecuteDBL</code> operation) <code><dynamic fetch statement></code> using this cursor will return, but any <code>ResSpec</code> specified with such a <code><dynamic fetch statement></code> overrides this <code>ResSpec</code> . The <i>n</i> th <code>SQLDataTypeDescriptor</code> of the <code>ResSpec</code> contains the description of the <i>n</i> th column of the table defined by the prepared statement associated with the <code><dynamic cursor name></code> referenced in the <code><dynamic open statement></code> .
10	If this statement contains a <code><using clause></code> that is a <code><using arguments></code> , then the RDA client sends an <code>ArgSpec</code> or <code>ResSpec</code> that contains one <code>SQLDataTypeDescriptor</code> for each <code><embedded variable name></code> in the <code><using clause></code> . The <i>n</i> th <code>SQLDataTypeDescriptor</code> corresponds to the <i>n</i> th <code><embedded variable name></code> . If this statement contains a <code><using clause></code> that is a <code><using descriptor></code> , then the RDA client sends an <code>ArgSpec</code> or <code>ResSpec</code> that contains one <code>SQLDataTypeDescriptor</code> for each item descriptor area in the SQL descriptor area referenced by the <code><using descriptor></code> . If this statement does not contain a <code><using clause></code> , then the RDA client does not send any <code>ArgSpec</code> or <code>ResSpec</code> .
11	No <code>ArgSpec</code> is sent by the RDA client, even if the <code><describe statement></code> contains a <code><using descriptor></code> clause whose <code><descriptor name></code> is a reference to an <code><embedded host variable></code> .

ASN.1 Module with X/Open Changes

Parts of Chapter 2 of this document specify changes to the ASN.1 module of the ISO/IEC RDA SQL standard. This appendix contains the resulting ASN.1 module. Change bars in the right margin identify the X/Open changes.

```

XOPEN-RDASQL { iso (1) national-member-body (2) uk (826) national (0)
                x-open (1050) data-management (3) x-open-rda (2)
                module (0) version-1 (1) }
--
-- *****
--
DEFINITIONS IMPLICIT TAGS ::= BEGIN
--
-- *****
--
IMPORTS
--
    AP-title,
    AE-qualifier,
    AP-invocation-identifier,
    AE-invocation-identifier
    FROM ACSE-1 { joint-iso-ccitt standard acse (8650) }
--
;

-- *****
--
-- RDA SQL APDUs
--

-- top-level APDU CHOICE

RDA-APDU ::= CHOICE
{
    r-Initialize-RI          [0] R-Initialize-RI,
    r-Initialize-RC          [1] R-Initialize-RC,
    r-Synchronize-RI        [2] R-Synchronize-RI,
    r-Terminate-RI          [3] R-Terminate-RI,
    r-Terminate-RC          [4] R-Terminate-RC,
    r-BeginTransaction-RI    [5] R-BeginTransaction-RI,
    r-BeginTransaction-RC    [6] R-BeginTransaction-RC,
    r-Commit-RI             [7] R-Commit-RI,
    r-Commit-RC             [8] R-Commit-RC,
    r-Rollback-RI           [9] R-Rollback-RI,
    r-Rollback-RC          [10] R-Rollback-RC,
    r-Cancel-RI             [11] R-Cancel-RI,
    r-Cancel-RC            [12] R-Cancel-RC,
    r-Status-RI            [13] R-Status-RI,
    r-Status-RC            [14] R-Status-RC,
    r-Open-RI              [15] R-Open-RI,
    r-Open-RC              [16] R-Open-RC,
    r-Close-RI             [17] R-Close-RI,
    r-Close-RC            [18] R-Close-RC,
    r-ExecuteDBL-RI        [19] R-ExecuteDBL-RI,
    r-ExecuteDBL-RC        [20] R-ExecuteDBL-RC,
    r-DefineDBL-RI         [21] R-DefineDBL-RI,
    r-DefineDBL-RC         [22] R-DefineDBL-RC,
    r-InvokeDBL-RI         [23] R-InvokeDBL-RI,
    r-InvokeDBL-RC         [24] R-InvokeDBL-RC,
    r-DropDBL-RI           [25] R-DropDBL-RI,
    r-DropDBL-RC           [26] R-DropDBL-RC
}

```

ASN.1 Module with X/Open Changes

```
-- individual APDU definitions.

R-Initialize-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Initialize-req
    [0] R-Initialize-Request
  }

R-Initialize-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Initialize-res
      [0] R-Initialize-Result,
      r-Initialize-err
      [1] R-Initialize-Error
    }
  }

R-Synchronize-RI ::= SEQUENCE
  {
  }

R-Terminate-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Terminate-req
    [0] R-Terminate-Request
  }

R-Terminate-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Terminate-res
      [0] R-Terminate-Result,
      r-Terminate-err
      [1] R-Terminate-Error
    }
  }

R-BeginTransaction-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-BeginTransaction-req
    [0] NULL
  }

R-BeginTransaction-RC ::= SEQUENCE
  { operationID
    OperationID,
    r-BeginTransaction-err
    [0] R-BeginTransaction-Error
  }

R-Commit-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Commit-req
    [0] NULL
  }

R-Commit-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Commit-res
      [0] R-Commit-Result,
      r-Commit-err
      [1] R-Commit-Error
    }
  }

R-Rollback-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Rollback-req
    [0] NULL
  }
```

```

R-Rollback-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Rollback-res
      [0] NULL,
      r-Rollback-err
      [1] R-Rollback-Error
    }
  }

R-Cancel-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Cancel-req
    [0] R-Cancel-Request
  }

R-Cancel-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Cancel-res
      [0] R-Cancel-Result,
      r-Cancel-err
      [1] R-Cancel-Error
    }
  }

R-Status-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Status-req
    [0] R-Status-Request
  }

R-Status-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Status-res
      [0] R-Status-Result,
      r-Status-err
      [1] R-Status-Error
    }
  }

R-Open-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Open-req
    [0] R-Open-Request
  }

R-Open-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Open-res
      [0] R-Open-Result,
      r-Open-err
      [1] R-Open-Error
    }
  }

R-Close-RI ::= SEQUENCE
  { operationID
    OperationID,
    r-Close-req
    [0] R-Close-Request
  }

R-Close-RC ::= SEQUENCE
  { operationID
    OperationID,
    res-or-err
    CHOICE
    { r-Close-res
      [0] R-Close-Result,
      r-Close-err
      [1] R-Close-Error
    }
  }

```

ASN.1 Module with X/Open Changes

```
R-ExecuteDBL-RI ::= SEQUENCE
  { operationID
    r-ExecuteDBL-req
  }

R-ExecuteDBL-RC ::= SEQUENCE
  { operationID
    res-or-err
    { r-ExecuteDBL-res
      r-ExecuteDBL-err
    }
  }

R-DefinedBL-RI ::= SEQUENCE
  { operationID
    r-DefinedBL-req
  }

R-DefinedBL-RC ::= SEQUENCE
  { operationID
    res-or-err
    { r-DefinedBL-res
      r-DefinedBL-err
    }
  }

R-InvokeDBL-RI ::= SEQUENCE
  { operationID
    r-InvokeDBL-req
  }

R-InvokeDBL-RC ::= SEQUENCE
  { operationID
    res-or-err
    { r-InvokeDBL-res
      r-InvokeDBL-err
    }
  }

R-DropDBL-RI ::= SEQUENCE
  { operationID
    r-DropDBL-req
  }

R-DropDBL-RC ::= SEQUENCE
  { operationID
    res-or-err
    { r-DropDBL-res
      r-DropDBL-err
    }
  }
```

```

-- *****

R-Initialize-Request      ::= SEQUENCE
  { dialogueIDSuffix      [0] DialogueIDSuffix,
    identityOfUser        [1] VisibleString,
    userAuthenticationData [2] AuthenticationData OPTIONAL,
    controlServiceDataRequested [3] BOOLEAN DEFAULT FALSE,
    functionalUnitsRequested [4] FunctionalUnits,
    sQLInitializeArgument [30] SQLInitializeArgument OPTIONAL
  }

R-Initialize-Result      ::= SEQUENCE
  { controlServiceData    [0] SEQUENCE
    { controlServicesAllowed [0] BOOLEAN DEFAULT TRUE,
      controlAuthenticationData [1] AuthenticationData OPTIONAL
    } OPTIONAL,
    functionalUnitsAllowed [1] FunctionalUnits,
    sQLInitializeResult    [30] SQLInitializeResult OPTIONAL
  }

R-Initialize-Error      ::= CHOICE
  { accessControlViolation AccessControlViolation,
    duplicateDialogueID    DuplicateDialogueID,
    invalidSequence        InvalidSequence,
    operationAborted       OperationAborted,
    userAuthenticationFailure UserAuthenticationFailure
  }

-- *****

R-Terminate-Request      ::= NULL

R-Terminate-Result      ::= NULL

R-Terminate-Error       ::= CHOICE
  { duplicateOperationID   DuplicateOperationID,
    invalidSequence        InvalidSequence,
    operationAborted       OperationAborted,
    serviceNotNegotiated   ServiceNotNegotiated
  }

-- *****

R-BeginTransaction-Error ::= CHOICE
  { duplicateOperationID   DuplicateOperationID,
    invalidSequence        InvalidSequence,
    operationAborted       OperationAborted,
    serviceNotNegotiated   ServiceNotNegotiated
  }

```


ASN.1 Module with X/Open Changes

```
-- *****

R-Commit-Result ::= SEQUENCE
  { transactionResult [0] ENUMERATED
    { committed (0),
      rolledBack (1)
    }
  }

R-Commit-Error ::= CHOICE
  { duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence
  }

-- *****

R-Rollback-Error ::= CHOICE
  { duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence
  }

-- *****

R-Cancel-Request ::= SEQUENCE
  { controlledDialogue [0] SEQUENCE
    { dialogueIDClientInvocation [0] DialogueIDClientInvocation
    OPTIONAL,
      dialogueIDSuffix [1] DialogueIDSuffix,
      controlAuthenticationData [2] AuthenticationData
    }
    OPTIONAL,
    listOfOperationID [1] SEQUENCE OF OperationID OPTIONAL
  }

R-Cancel-Result ::= NULL

R-Cancel-Error ::= CHOICE
  { controlAuthenticationFailure ControlAuthenticationFailure,
    controlServicesNotAllowed ControlServicesNotAllowed,
    dialogueIDUnknown DialogueIDUnknown,
    duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    operationAborted OperationAborted,
    serviceNotNegotiated ServiceNotNegotiated
  }
```

```

-- *****

R-Status-Request ::= SEQUENCE
  { controlledDialogue [0] SEQUENCE
    { dialogueIDClientInvocation [0] DialogueIDClientInvocation
OPTIONAL,
    dialogueIDSuffix [1] DialogueIDSuffix,
    controlAuthenticationData [2] AuthenticationData
    }
    listOfOperationID [1] SEQUENCE OF OperationID OPTIONAL
  }

R-Status-Result ::= SEQUENCE
  { listOfStatusInformation [0] SEQUENCE OF StatusInformation OPTIONAL
  }

R-Status-Error ::= CHOICE
  { controlAuthenticationFailure ControlAuthenticationFailure,
    controlServicesNotAllowed ControlServicesNotAllowed,
    dialogueIDUnknown DialogueIDUnknown,
    duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    operationAborted OperationAborted,
    serviceNotNegotiated ServiceNotNegotiated
  }

-- *****

R-Open-Request ::= SEQUENCE
  { dataResourceHandle [0] DataResourceHandle,
    dataResourceName [2] VisibleString OPTIONAL,
    sqlAccessControlData [3] AccessControlData OPTIONAL,
    sqlUsageMode [4] SQLUsageMode DEFAULT retrieval,
    sqlOpenArgument [30] SQLOpenArgument OPTIONAL
  }

R-Open-Result ::= SEQUENCE
  { sqlOpenResult [30] SQLOpenResult OPTIONAL
  }

R-Open-Error ::= CHOICE
  { dataResourceAlreadyOpen DataResourceAlreadyOpen,
    dataResourceNameNotSpecified DataResourceNameNotSpecified,
    dataResourceNotAvailable DataResourceNotAvailable,
    dataResourceUnknown DataResourceUnknown,
    duplicateDataResourceHandle DuplicateDataResourceHandle,
    duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    operationAborted OperationAborted,
    operationCancelled OperationCancelled,
    serviceNotNegotiated ServiceNotNegotiated,
    sqlOpenError SQLOpenError
  }

```

ASN.1 Module with X/Open Changes

```
-- *****

R-Close-Request ::= SEQUENCE
  { listOfDataResourceHandle [0] SEQUENCE OF DataResourceHandle OPTIONAL
  }

R-Close-Result ::= SEQUENCE
  { listOfCloseExceptions [0] SEQUENCE OF CloseException OPTIONAL
  }

R-Close-Error ::= CHOICE
  { duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    operationAborted OperationAborted,
    operationCancelled OperationCancelled,
    serviceNotNegotiated ServiceNotNegotiated,
    sqlCloseError SQLCloseError
  }

-- *****

R-ExecutedDBL-Request ::= SEQUENCE
  { dataResourceHandle [0] DataResourceHandle OPTIONAL,
    sqlDBLStatement [1] SQLDBLStatement,
    sqlDBLArgumentSpecification [2] SQLDBLArgumentSpecification OPTIONAL,
    sqlDBLResultSpecification [3] SQLDBLResultSpecification OPTIONAL,
    dblArguments CHOICE
    { singleArgument [4] SEQUENCE
      { repetitionCount [0] INTEGER DEFAULT 1,
        sqlDBLArgumentValues [1] SQLDBLArgumentValues OPTIONAL
      },
      multipleArgument [5] SEQUENCE
      { listOfSQLDBLArgumentValues [0] SEQUENCE OF SQLDBLArgumentValues
      }
    }
    returnsSQLDiagnostics [6] BOOLEAN DEFAULT TRUE
  }

R-ExecutedDBL-Result ::= SEQUENCE
  { sqlDBLResultSpecification [1] SQLDBLResultSpecification OPTIONAL,
    listOfResultValues [2] SEQUENCE OF ResultValues OPTIONAL
  }

R-ExecutedDBL-Error ::= CHOICE
  { badRepetitionCount BadRepetitionCount,
    dataResourceHandleNotSpecified DataResourceHandleNotSpecified,
    dataResourceHandleUnknown DataResourceHandleUnknown,
    duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    noDataResourceAvailable NoDataResourceAvailable,
    operationAborted OperationAborted,
    operationCancelled OperationCancelled,
    serviceNotNegotiated ServiceNotNegotiated,
    transactionRolledBack TransactionRolledBack,
    sqlExecuteDBLError SQLExecuteDBLError
  }
```

```

-- *****

R-DefinedDBL-Request ::= SEQUENCE
{
  commandHandle          [0] CommandHandle,
  dataResourceHandle     [1] DataResourceHandle OPTIONAL,
  SQLDBLStatement       [2] SQLDBLStatement,
  SQLDBLArgumentSpecification [3] SQLDBLArgumentSpecification OPTIONAL,
  SQLDBLResultSpecification [4] SQLDBLResultSpecification OPTIONAL
}

R-DefinedDBL-Result ::= SEQUENCE
{
  SQLDBLResultSpecification [0] SQLDBLResultSpecification OPTIONAL,
  SQLDBLException          [1] SQLDBLException OPTIONAL
}

R-DefinedDBL-Error ::= CHOICE
{
  dataResourceHandleNotSpecified DataResourceHandleNotSpecified,
  dataResourceHandleUnknown      DataResourceHandleUnknown,
  duplicateCommandHandle         DuplicateCommandHandle,
  duplicateOperationID           DuplicateOperationID,
  invalidSequence                InvalidSequence,
  noDataResourceAvailable       NoDataResourceAvailable,
  operationAborted               OperationAborted,
  operationCancelled             OperationCancelled,
  serviceNotNegotiated           ServiceNotNegotiated,
  SQLDefineDBLError             SQLDefineDBLError
}

-- *****

R-InvokeDBL-Request ::= SEQUENCE
{
  commandHandle          [0] CommandHandle,
  dblArguments           CHOICE
  {
    singleArgument      [1] SEQUENCE
    {
      repetitionCount [0] INTEGER DEFAULT 1,
      SQLDBLArgumentValues [1] SQLDBLArgumentValues OPTIONAL
    },
    multipleArgument    [2] SEQUENCE
    {
      listOfSQLDBLArgumentValues [0] SEQUENCE OF SQLDBLArgumentValues
    }
  }
  OPTIONAL
}

R-InvokeDBL-Result ::= SEQUENCE
{
  SQLDBLResultSpecification [0] SQLDBLResultSpecification OPTIONAL,
  listOfResultValues        [2] SEQUENCE OF ResultValues OPTIONAL
}

R-InvokeDBL-Error ::= CHOICE
{
  badRepetitionCount      BadRepetitionCount,
  commandHandleUnknown    CommandHandleUnknown,
  duplicateOperationID     DuplicateOperationID,
  invalidSequence         InvalidSequence,
  operationAborted        OperationAborted,
  operationCancelled      OperationCancelled,
  serviceNotNegotiated    ServiceNotNegotiated,
  transactionRolledBack   TransactionRolledBack,
  SQLInvokeDBLError       SQLInvokeDBLError
}

```

ASN.1 Module with X/Open Changes

```
-- *****
R-DropDBL-Request ::= SEQUENCE
  { listOfCommandHandle [0] SEQUENCE OF CommandHandle OPTIONAL
  }

R-DropDBL-Result ::= SEQUENCE
  { listOfDropDBLExceptions [0] SEQUENCE OF DropDBLException OPTIONAL
  }

R-DropDBL-Error ::= CHOICE
  { duplicateOperationID DuplicateOperationID,
    invalidSequence InvalidSequence,
    operationAborted OperationAborted,
    operationCancelled OperationCancelled,
    serviceNotNegotiated ServiceNotNegotiated
  }

-- *****
--
-- Definitions of common types, ordered alphabetically
--

AuthenticationData ::= CHOICE
  { cstring [0] IA5String,
    ostring [1] OCTET STRING,
    bstring [2] BIT STRING
  }

CloseException ::= SEQUENCE
  { dataResourceHandle [0] DataResourceHandle,
    closeException CHOICE
    { dataResourceHandleUnknown [1] NULL
    }
  }

CommandHandle ::= INTEGER

DataResourceHandle ::= INTEGER

DialogueIDClientInvocation ::= SEQUENCE
  { apTitle [0] AP-title,
    aeQualifier [1] AE-qualifier,
    apInvocationID [2] AP-invocation-identifier,
    aeInvocationID [3] AE-invocation-identifier
  }

DialogueIDSuffix ::= CHOICE
  { ostring [0] OCTET STRING
  }

DropDBLException ::= SEQUENCE
  { commandHandle [0] CommandHandle,
    dropDBLException CHOICE
    { commandHandleUnknown [1] NULL
    }
  }
```

```

FunctionalUnits ::= BIT STRING
  { termination (0),
    transaction (1),
    cancel (2),
    status (3),
    resource (4),
    immediate-DBL (5),
    stored-DBL (6)
  }

OperationID ::= INTEGER

ResultValues ::= SEQUENCE
  { SQLDBLException [0] SQLDBLException,
    SQLDBLResultValues [1] SQLDBLResultValues OPTIONAL
  }

StatusInformation ::= SEQUENCE
  { operationID [0] OperationID,
    operationStatus CHOICE
    { operationIDUnknown [1] NULL,
      awaitingExecution [2] NULL,
      executing [3] NULL,
      finished [4] NULL,
      cancelled [5] NULL,
      aborted [6] VisibleString
    }
  }

-- *****

```

ASN.1 Module with X/Open Changes

```
--
-- Definitions of generic ASN.1 errors
--

AccessControlViolation          ::= [APPLICATION 0] NULL

BadRepetitionCount              ::= [APPLICATION 1] NULL

CommandHandleUnknown            ::= [APPLICATION 2] NULL

ControlAuthenticationFailure    ::= [APPLICATION 3] NULL

ControlServicesNotAllowed       ::= [APPLICATION 4] NULL

DataResourceAlreadyOpen        ::= [APPLICATION 5] SEQUENCE
  { dataResourceHandle          [0] DataResourceHandle
  }

DataResourceHandleNotSpecified ::= [APPLICATION 6] NULL

DataResourceHandleUnknown       ::= [APPLICATION 7] NULL

DataResourceNameNotSpecified    ::= [APPLICATION 8] NULL

DataResourceNotAvailable        ::= [APPLICATION 9] ErrorDiagnostic

DataResourceUnknown             ::= [APPLICATION 10] NULL

DialogueIDUnknown               ::= [APPLICATION 11] NULL

DuplicateCommandHandle          ::= [APPLICATION 12] NULL

DuplicateDataResourceHandle     ::= [APPLICATION 13] NULL

DuplicateDialogueID             ::= [APPLICATION 14] NULL

DuplicateOperationID            ::= [APPLICATION 15] NULL

InvalidSequence                 ::= [APPLICATION 16] SEQUENCE
  { diagnosticInformation       [0] ENUMERATED
    { dialogueNotActive         (1),
      dialogueInitializing      (2),
      dialogueAlreadyActive     (3),
      transactionNotOpen        (4),
      transactionOpen           (5),
      transactionTerminating    (6),
      dialogueTerminating       (7)
    }
  } OPTIONAL

NoDataResourceAvailable         ::= [APPLICATION 17] NULL

OperationAborted                ::= [APPLICATION 18] ErrorDiagnostic

OperationCancelled              ::= [APPLICATION 19] NULL

ServiceNotNegotiated           ::= [APPLICATION 20] NULL

TransactionRolledBack           ::= [APPLICATION 21] NULL
```

```
UserAuthenticationFailure      ::= [APPLICATION 22] NULL

ErrorDiagnostic                 ::= SEQUENCE
{
  errorType                     [0] ENUMERATED
    { transient
      permanent
    } DEFAULT transient,
  diagnosticInformation          [1] VisibleString OPTIONAL
}

-- *****
```


ASN.1 Module with X/Open Changes

```
--
-- Definitions of SQL Specialization parameters
--

AccessControlData ::= CHOICE
{
  cstring      [0] IA5String,
  ostring      [1] OCTET STRING,
  bstring      [2] BIT STRING
}

ExceptionInfo ::= SEQUENCE
{
  returnedSQLSTATE [0] VisibleString,
  classOrigin      [1] VisibleString,
  subclassOrigin   [2] VisibleString,
  messageText      [3] VisibleString OPTIONAL
}

SQLDataTypeDescriptor ::= SEQUENCE
{
  nullable      [0] BOOLEAN DEFAULT TRUE,
  colName       [1] VisibleString OPTIONAL,
  typeDescriptor CHOICE
  {
    characterType [5] SEQUENCE
    -- SQL Type: character
    {
      charSet      OBJECT IDENTIFIER OPTIONAL,
      length       INTEGER,
      fixedLengthEncoding BOOLEAN
    },
    numericType   [6] SEQUENCE
    -- SQL Type: numeric
    {
      precision    INTEGER,
      scale        INTEGER
    },
    decimalType   [7] SEQUENCE
    -- SQL Type: decimal
    {
      precision    INTEGER,
      scale        INTEGER
    },
    integerType   [8] SEQUENCE
    -- SQL Type: integer
    {
      precision    INTEGER,
      precisionBase ENUMERATED
      {
        binary      (0),
        decimal     (1)
      }
    },
    smallIntType  [9] SEQUENCE
    -- SQL Type: smallInt
    {
      precision    INTEGER,
      precisionBase ENUMERATED
      {
        binary      (0),
        decimal     (1)
      }
    },
    floatType     [10] SEQUENCE
    -- SQL Type: float
    {
      mantissaPrecision INTEGER,
      maxExponent       INTEGER
    },
    realType      [11] SEQUENCE
  }
}
```

```

-- SQL Type: real
{ mantissaPrecision          INTEGER,
  maxExponent                INTEGER
},
doublePrecisionType         [12] SEQUENCE
-- SQL Type: doublePrecision
{ mantissaPrecision          INTEGER,
  maxExponent                INTEGER
},
varcharType                 [15] SEQUENCE
-- SQL Type: varchar
{ charSet                    OBJECT IDENTIFIER OPTIONAL,
  length                      INTEGER
}
}

SQLDBLArgumentSpecification ::= SEQUENCE
{ listOfSQLDataTypeDescriptor [0] SEQUENCE OF SQLDataTypeDescriptor
}

SQLDBLArgumentValues        ::= SQLValueList

SQLDBLException             ::= SEQUENCE
{ SQLSTATE                   [0] VisibleString OPTIONAL,
  SQLCODE                     [1] INTEGER OPTIONAL,
  SQLErrorText                [2] VisibleString OPTIONAL,
  SQLDiagnostics              [3] SQLDiagnostics OPTIONAL
}

SQLDBLResultSpecification   ::= SEQUENCE
{ listOfSQLDataTypeDescriptor [0] SEQUENCE OF SQLDataTypeDescriptor
}

SQLDBLResultValues         ::= SQLValueList

SQLDBLStatement             ::= SEQUENCE
{ statementText              [0] OCTET STRING,
  charSet                     [1] OBJECT IDENTIFIER OPTIONAL
}

SQLDiagnostics              ::= SEQUENCE
{ rowCount                    [0] INTEGER OPTIONAL,
  exceptionList               [3] SEQUENCE OF ExceptionInfo
}

SQLInitializeArgument       ::= SEQUENCE
{ SQLConformanceLevelDefault [0] OBJECT IDENTIFIER OPTIONAL,
  userData                    [1] OCTET STRING OPTIONAL
}

SQLInitializeResult         ::= SEQUENCE
{ userData                     [0] OCTET STRING OPTIONAL
}

SQLOpenArgument             ::= SEQUENCE
{ charSet                     [0] OBJECT IDENTIFIER OPTIONAL,
  SQLConformanceLevel        [1] OBJECT IDENTIFIER OPTIONAL,
  SQLDiagnosticsRequested     [2] ENUMERATED
}

```

ASN.1 Module with X/Open Changes

```

    { always                (0),
      onRequest             (1),
      never                 (2)
    } DEFAULT never
  }

SQLOpenResult ::= SEQUENCE
  { charSet                [0] OBJECT IDENTIFIER OPTIONAL,
    charSetNotSupported    [1] BOOLEAN DEFAULT FALSE,
    sqlConformanceLevel    [2] OBJECT IDENTIFIER OPTIONAL
  }

SQLUsageMode ::= ENUMERATED
  { retrieval              (0),
    update                  (1)
  }

SQLValue ::= SEQUENCE
  { dataItem               CHOICE
    { characterItem        [0] OCTET STRING,
      numericItem          [1] INTEGER,
      decimalItem          [2] INTEGER,
      integerItem          [3] INTEGER,
      smallIntItem         [4] INTEGER,
      floatItem            [5] REAL,
      realItem             [6] REAL,
      doublePrecisionItem  [7] REAL,
      varcharItem          [10] OCTET STRING
    }
    indicator              [30] INTEGER OPTIONAL
  }

SQLValueList ::= CHOICE
  { listOfSQLValue        [1] SEQUENCE OF SQLValue
  }

-- *****
```

```

--
-- Definitions of RDA SQL Specialization Errors
--

HostIdentifierError          ::= [APPLICATION 23] NULL

InvalidSQLConformanceLevel  ::= [APPLICATION 24] NULL

RDATransactionNotOpen      ::= [APPLICATION 25] NULL

RDATransactionOpen         ::= [APPLICATION 26] NULL

SQLAccessControlViolation   ::= [APPLICATION 27] NULL

SQLDatabaseResourceAlreadyOpen ::= [APPLICATION 28] NULL

SQLDBLArgumentCountMismatch ::= [APPLICATION 29] NULL

SQLDBLArgumentTypeMismatch  ::= [APPLICATION 30] NULL

SQLDBLNoCharSet             ::= [APPLICATION 31] NULL

SQLDBLStatementNotAllowed   ::= [APPLICATION 32] NULL

SQLUsageModeViolation       ::= [APPLICATION 33] NULL

--
-- SQL Specialization errors which can be returned for specific RDA operations
--

SQLCloseError               ::= CHOICE
  { rDATransactionOpen      RDATransactionOpen
  }

SQLDefinedDBLError          ::= CHOICE
  { hostIdentifierError      HostIdentifierError,
    SQLDBLNoCharSet         SQLDBLNoCharSet,
    SQLDBLStatementNotAllowed SQLDBLStatementNotAllowed,
    SQLUsageModeViolation   SQLUsageModeViolation
  }

SQLExecutedDBLError         ::= CHOICE
  { hostIdentifierError      HostIdentifierError,
    rDATransactionNotOpen   RDATransactionNotOpen,
    SQLDBLArgumentCountMismatch SQLDBLArgumentCountMismatch,
    SQLDBLArgumentTypeMismatch SQLDBLArgumentTypeMismatch,
    SQLDBLNoCharSet         SQLDBLNoCharSet,
    SQLDBLStatementNotAllowed SQLDBLStatementNotAllowed,
    SQLUsageModeViolation   SQLUsageModeViolation
  }

```

ASN.1 Module with X/Open Changes

```
SQLInvokeDBLError ::= CHOICE
  { rDATransactionNotOpen      RDATransactionNotOpen,
    SQLDBLArgumentCountMismatch SQLDBLArgumentCountMismatch,
    SQLDBLArgumentTypeMismatch SQLDBLArgumentTypeMismatch,
    SQLUsageModeViolation      SQLUsageModeViolation
  }

SQLOpenError ::= CHOICE
  { invalidSQLConformanceLevel InvalidSQLConformanceLevel,
    rDATransactionOpen         RDATransactionOpen,
    SQLAccessControlViolation  SQLAccessControlViolation,
    SQLDatabaseResourceAlreadyOpen SQLDatabaseResourceAlreadyOpen
  }

-- *****
END          -- RDA SQL ASN.1 Module
-- *****
```


Glossary

ACSE

Association control service element.

APDU

Application protocol data unit.

ASN.1

(Abstract Syntax Notation One) A notation, defined in ISO 8824:1990, that allows data to be described in a machine-independent fashion.

association control service element

The application service element responsible for association establishment and release. All OSI application entities contain an ACSE.

application protocol data unit

A unit of data specified in an application layer protocol and consisting of application protocol control information and possibly application user data.

byte

An individually addressable unit of data storage that is equal to or larger than an octet.

client

(RDA Client) The RDA service-user that initialises an RDA dialogue and requests database access from a remote database server.

database

(SQL Database Resource) The data and the schemas describing it, as defined in the X/Open SQL specification.

database server

An application process that supplies database storage facilities and provides, through OSI communication, database services to other application processes called RDA clients.

dynamic SQL

A way to execute SQL statements whose specifics may not be known at compile time.

octet

Eight bits in a row, not necessarily at an addressable machine boundary.

presentation layer

Layer six of the OSI Reference Model; the presentation layer provides the mechanisms for negotiating the common method for representing information and for transferring information so that the semantics are preserved during the transfer.

server

(RDA server) The RDA service-user within a database server that provides database access to remote RDA clients.

session layer

Layer five of the OSI Reference Model; the session layer provides the mechanisms for establishing a session connection between cooperating users and for the organised and synchronised exchange of data between those users.

SQL

(Structured Query Language) A database language widely accepted as an interface to relational database management systems.

Index

<allocate descriptor statement>	4, 10
<alter table statement>	12
<commit statement>	4
<connect statement>	4, 11
<create index statement>	3, 12
<deallocate descriptor statement>	4, 10
<describe statement>	10
<disconnect statement>	4, 11
<drop index statement>	3, 12
<drop table statement>	12
<drop view statement>	12
<dynamic close statement>	10
<dynamic declare cursor>	10
<dynamic delete statement: positioned>	10
<dynamic fetch statement>	10
<dynamic open statement>	10
<dynamic update statement: positioned>	10
<embedded variable name>	10
<execute immediate statement>	10
<execute statement>	10
<get descriptor statement>	4, 10
<get diagnostics statement>	4-6
<prepare statement>	10
<revoke statement>	12
<rollback statement>	4
<schema definition>	12
<set connection statement>	4, 11
<set descriptor statement>	4, 10
<table definition>	3
<update statement: positioned>	3
<view definition>	3
ACSE	14, 59
APDU	14, 59
application context	13
Application Context Name	33
Application Context Object Identifier	14
application protocol data unit	59
ASN.1	59
ASN.1 module	3, 7-9, 15, 33
X/Open changes	39
association control service element	59
byte	59
Called Presentation Selector	14
Called Session Selector	14
Calling Presentation Selector	14
Calling Session Selector	14
character set	8, 13
charSet	8, 33
client	59
client, RDA	1
control services	13
create-index-statement	3
create-table-statement	3
create-view-statement	3
Data Variable	8
database	59
database server	59
dataItem	9
dblArguments	7
default contexts	14
descriptor area	10
diagnostics management	6
doublePrecisionItem	8
drop-index-statement	3
dynamic SQL	59
entry level	3
Error Parameters	4
fully-encoded-data	14
host variable	10
implementation agreements	1, 13
INTEGER	
magnitude	15
ISO/IEC RDA Generic standard	1, 3
ISO/IEC RDA SQL standard	1, 3
length	8
limits	
common parameters	30
listOfSQLDBLArgumentValues	5, 34
maximum values	13
Mode	14
object identifiers	33
other authorities	33
X/Open	33
octet	59
opened data resource entity	6
operation limits	13
P-CONNECT	14
parameters	15
ignored	15
limitations	15
mandatory	15
optional	15

R-BeginTransaction error response	19	operationID.....	19
R-BeginTransaction request.....	19	serviceNotNegotiated	19
R-Cancel error response.....	22	R-BeginTransaction request	
R-Cancel request.....	22	operationID.....	19
R-Cancel result response	22	R-Cancel	22
R-Close error response	27	R-Cancel error response	
R-Close request.....	27	controlAuthenticationFailure	22
R-Close result response.....	27	controlServicesNotAllowed.....	22
R-Commit error response	20	diagnosticInformation.....	22
R-Commit request.....	20	dialogueIDUnknown.....	22
R-Commit result response.....	20	duplicateOperationID	22
R-ExecuteDBL error response.....	29	errorType	22
R-ExecuteDBL request.....	28	invalidSequence.....	22
R-ExecuteDBL result response	28	operationAborted.....	22
R-Initialize error response	17	operationID.....	22
R-Initialize request.....	16	serviceNotNegotiated	22
R-Initialize result response.....	17	R-Cancel request	
R-Open error response	26	aE-invocationID	22
R-Open request.....	25	aE-qualifier.....	22
R-Open result response.....	25	aP-invocationID	22
R-Rollback error response	21	aP-title	22
R-Rollback request.....	21	controlAuthenticationData	22
R-Rollback result response.....	21	controlledDialogue	22
R-Status error response	24	dialogueID	22
R-Status request.....	23	dialogueIDClientInvocation	22
R-Status result response.....	23	dialogueIDSuffix	22
R-Terminate error response.....	18	listOfOperationID	22
R-Terminate request.....	18	OperationID.....	22
R-Terminate result response	18	operationID.....	22
SQLDataTypeDescriptor.....	31	R-Cancel result response	
SQLDBLException	31	operationID.....	22
SQLValue.....	32	R-Close.....	27
unused	15	R-Close error response	
usage	13, 15, 34	diagnosticInformation.....	27
values	15	duplicateOperationID	27
X/Open extensions	15	errorType	27
Presentation Address.....	14	invalidSequence.....	27
Presentation Context Definition List.....	14	operationAborted.....	27
presentation layer.....	14, 59	operationCancelled	27
Presentation Requirements.....	14	operationID.....	27
Presentation Service	14	rDATransactionOpen.....	27
Presentation user data	14	serviceNotNegotiated	27
Presentation Version	14	SQLCloseError.....	27
protocol stack	14	R-Close request	
R-BeginTransaction	19	DataResourceHandle.....	27
R-BeginTransaction error response		listOfDataResourceHandle	27
diagnosticInformation.....	19	operationID.....	27
duplicateOperationID	19	R-Close result response	
errorType	19	CloseException	27
invalidSequence.....	19	closeException	27
operationAborted.....	19	dataResourceHandle	27

Index

dataResourceHandleUnknown.....	27	SQLDBLArgumentSpecification.....	28
listOfCloseExceptions	27	SQLDBLArgumentValues	28
operationID.....	27	SQLDBLArgumentValues.....	28
R-Commit.....	20	SQLDBLResultSpecification	28
R-Commit error response		SQLDBLStatement.....	28
diagnosticInformation.....	20	statementText	28
duplicateOperationID	20	R-ExecuteDBL result response	
invalidSequence.....	20	listOfResultValues.....	28
operationID.....	20	operationID.....	28
R-Commit request		ResultValues	28
operationID.....	20	SQLDBLException.....	28
R-Commit result response		SQLDBLResultSpecification	28
operationID.....	20	SQLDBLResultValues	28
transactionResult	20	R-ExecuteDBL service.....	6
R-DefineDBL.....	13	R-ExecuteDBL-Request	7
R-DropDBL.....	13	R-ExecuteDBLService	4
R-ExecuteDBL.....	3-5, 10-12, 28	R-Initialize	16
result parameters.....	6	R-Initialize error response	
use of SQL argument	6	accessControlViolation	17
R-ExecuteDBL error response		diagnosticInformation.....	17
badRepetitionCount	29	duplicateDialogueID	17
dataResourceHandleNotSpecified	29	errorType	17
dataResourceHandleUnknown.....	29	invalidSequence.....	17
diagnosticInformation.....	29	operationAborted.....	17
duplicateOperationID	29	operationID.....	17
errorType	29	userAuthenticationFailure.....	17
hostIdentifierError.....	29	R-Initialize request	
invalidSequence.....	29	aE-invocationID	16
noDataResourceAvailable	29	aE-qualifier.....	16
operationAborted.....	29	aP-invocationID	16
operationCancelled	29	aP-title	16
operationID.....	29	controlServiceDataRequested.....	16
rDATransactionNotOpen.....	29	dialogueID	16
serviceNotNegotiated	29	dialogueIDClientInvocation	16
SQLDBLArgumentCountMismatch.....	29	dialogueIDSuffix	16
SQLDBLArgumentTypeMismatch.....	29	functionalUnitsRequested	16
SQLDBLNoCharSet.....	29	identityOfUser	16
SQLDBLStatementNotAllowed.....	29	operationID.....	16
SQLExecuteDBLError	29	SQLConformanceLevelDefault	16
SQLUsageModeViolation	29	SQLInitializeArgument	16
transactionRolledBack.....	29	Stored Execution DBL	16
R-ExecuteDBL request		userAuthenticationData.....	16
charSet	28	userData	16
dataResourceHandle	28	R-Initialize result response	
dBLArguments.....	28	controlAuthenticationData	17
listOfSQLDBLArgumentValues	28	controlServiceData	17
multipleArguments	28	controlServicesAllowed	17
operationID.....	28	functionalUnitsAllowed	17
repetitionCount	28	operationID.....	17
returnSQLDiagnostics	28	SQLInitializeResult.....	17
singleArgument.....	28	userData	17

R-InvokeDBL.....	13	R-Status error response	
R-Open	5-6, 8, 25	controlAuthenticationFailure	24
R-Open error response		controlServicesNotAllowed	24
dataResourceAlreadyOpen.....	26	diagnosticInformation.....	24
dataResourceHandle	26	dialogueIDUnknown.....	24
dataResourceNameNotSpecified.....	26	duplicateOperationID	24
dataResourceNotAvailable	26	errorType	24
dataResourceUnknown.....	26	invalidSequence.....	24
diagnosticInformation.....	26	operationAborted.....	24
duplicateDataResourceHandle	26	operationID.....	24
duplicateOperationID	26	serviceNotNegotiated"	24"
errorType	26	R-Status request	
invalidSequence.....	26	aE-invocationID.....	23
invalidSQLConformanceLevel.....	26	aE-qualifier.....	23
operationAborted.....	26	aP-invocationID.....	23
operationCancelled.....	26	aP-title.....	23
operationID.....	26	controlAuthenticationData	23
rDATransactionOpen.....	26	controlledDialogue	23
serviceNotNegotiated	26	dialogueID	23
sQLAccessControlViolation.....	26	dialogueIDClientInvocation~	23
sQLDatabaseResourceAlreadyOpen	26	dialogueIDSuffix	23
sQLOpenError.....	26	listOfOperationID	23
R-Open request		OperationID.....	23
charSet	25	operationID.....	23
dataResourceHandle	25	R-Status result response	
dataResourceName	25	aborted.....	23
operationID.....	25	awaitingExecution	23
sQLAccessControlData	25	cancelled.....	23
sQLConformanceLevel	25	executing	23
sQLDiagnosticsRequested.....	25	finished	23
sQLOpenArgument	25	listOfStatusInformation	23
sQLUsageMode	25	operationID.....	23
R-Open result response		operationIDUnknown	23
charSet	25	operationStatus.....	23
charSetNotSupported.....	25	StatusInformation	23
operationID.....	25	R-Synchronize	18
sQLConformanceLevel	25	R-Terminate	18
sQLOpenResult.....	25	R-Terminate error response	
R-Open service	6	diagnosticInformation.....	18
R-Rollback	21	duplicateOperationID	18
R-Rollback error response		errorType	18
diagnosticInformation.....	21	invalidSequence.....	18
duplicateOperationID	21	operationAborted.....	18
invalidSequence.....	21	operationID.....	18
operationID.....	21	serviceNotNegotiated	18
R-Rollback request		R-Terminate request	
operationID.....	21	operationID.....	18
R-Rollback result response		R-Terminate result response	
operationID.....	21	operationID.....	18
R-Status.....	23	RDA Abstract Syntax.....	33

Index

RDA operation	
outstanding.....	13
pending.....	13
rejecting.....	13
RDA standards.....	1
repetitionCount.....	34
returnSQLDiagnostics.....	5
server.....	59
server, RDA.....	1
session layer.....	14, 59
Session Requirements.....	14
Session Version.....	14
Single-ASN1-type.....	14
SQL.....	60
SQL database resource.....	8
SQL diagnostics information.....	5
SQL Specific Service Parameters.....	5
sQLConformanceLevel.....	5, 33
SQLDataTypeDescriptor.....	8
characterType.....	31
charSet.....	31
colName.....	31
decimalType.....	31
doublePrecisionType.....	31
fixedLengthEncoding.....	31
floatType.....	31
integerType.....	31
length.....	31
mantissaPrecision.....	31
mantissaPrecison.....	31
maxExponent.....	31
nullable.....	31
numericType.....	31
precision.....	31
precisionBase.....	31
realType.....	31
scale.....	31
smallIntType.....	31
typeDescriptor.....	31
varCharType.....	31
SQLDataTypeDescriptors.....	10
sQLDBLArgumentSpecification.....	8
sQLDBLArgumentValues.....	8
sQLDBLException.....	5-6
SQLDBLException.....	7
classOrigin.....	31
exceptionInfo.....	31
exceptionList.....	31
messageText.....	31
returnedSQLState.....	31
rowCount.....	31
sQLCode.....	31
sQLDiagnostics.....	31
sQLErrorText.....	31
sQLState.....	31
subclassOrigin.....	31
sQLDBLResultSpecification.....	8
sQLDBLResultValues.....	8
sQLDBLStatement.....	4
SQLDBLStatementNotAllowed.....	4
sQLDBLStatementNotAllowed.....	4, 11
SQLDBLTransactionStatementNotAllowed.....	4
sQLDBLTransactionStatementNotAllowed.....	4
SQLDiagnostics.....	6
sQLDiagnostics.....	5-6
sQLDiagnosticsRequested.....	5-6
always.....	5-6
false.....	5
never.....	5-6
onRequest.....	5-6
true.....	5-6
sQLErrorText.....	5-6
sQLOpenArgument.....	5
SQLOpenArgument.....	7
SQLValue.....	9
characterItem.....	32
dataItem.....	32
decimalItem.....	32
doublePrecisionItem.....	32
floatItem.....	32
indicator.....	32
integerItem.....	32
numericItem.....	32
realItem.....	32
smallIntItem.....	32
varcharItem.....	32
statements	
connection management.....	11
data definition.....	12
dynamic SQL.....	10
SQL descriptor.....	10
transaction management.....	4
subparameters.....	15
syntactic element.....	3
transaction.....	13
typeDescriptor.....	8
typography, ISO.....	3
update-statement-positioned.....	3
User-data.....	14
VARCHAR.....	8
varcharItem.....	8
varcharType.....	8

