



INTERNATIONAL STANDARD ISO/IEC 9075-13:2003
TECHNICAL CORRIGENDUM 1

Published 2005-11-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Database languages — SQL —
Part 13:
SQL Routines and Types Using the Java™ Programming
Language (SQL/JRT)

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de base de données — SQL —

Partie 13: Routines et types utilisant le langage de programmation Java™ (SQL/JRT)

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9075-13:2003 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

Statement of purpose for rationale

A statement indicating the rationale for each change to ISO/IEC 9075-13:2003 is included. This is to inform the users of ISO/IEC 9075-13:2003 why it was judged necessary to change the original wording. In many cases, the reason is editorial or to clarify the wording; in some cases, it is to correct an error or an omission in the original wording.

Notes on numbering

Where this Technical Corrigendum introduces new Syntax, Access, General, and Conformance Rules, the new rules have been numbered as follows:

Rules inserted between, for example, Rules 7) and 8) are numbered 7.1), 7.2), etc. [or 7)a.1), 7)a.2), etc.]. Those inserted before Rule 1) are numbered 0.1), 0.2, etc.

Where this Technical Corrigendum introduces new subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, 4.3.2 and 4.3.3 are numbered 4.3.2a, 4.3.2b, etc. Those inserted before, for example, 4.3.1 are numbered 4.3.0, 4.3.0a, etc.

ICS 35.060

Ref. No. ISO/IEC 9075-13:2003/Cor.1:2005(E)

© ISO/IEC 2005 – All rights reserved

Published in Switzerland

AVA Documents
Document Sharing Hub

Contents

Page

Foreword.....	1
4 Concepts.....	1
4.3 Java class name resolution.....	1
4.6 Unhandled Java exceptions.....	2
4.10 Privileges.....	2
8 Additional common elements.....	2
8.2 <SQL Java path>.....	2
8.3 <routine invocation>.....	3
9 Schema definition and manipulation.....	3
9.4 <user-defined type definition>.....	3
9.8 <SQL-invoked routine>.....	4
10 Access control.....	4
10.3 <revoke statement>.....	4
11 Built-in procedures.....	5
11.1 SQLJ.INSTALL_JAR procedure.....	5
11.3 SQLJ.REMOVE_JAR procedure.....	5
11.4 SQLJ.ALTER_JAVA_PATH procedure.....	6
13 Information Schema.....	7
13.2 JARS view.....	7
14 Definition Schema.....	8
14.2 JARS base table.....	8
15 Status codes.....	9
15.1 Class and subclass values for uncaught Java exceptions.....	9
15.2 SQLSTATE.....	9
Annex B Implementation-defined elements.....	10
Annex E Routines tutorial.....	10
E.21 Exceptions.....	10

Tables

Table	Page
2 SQLSTATE class and subclass values.	9

Information technology — Database languages — SQL —

Part 13:

SQL Routines and Types Using the Java™ Programming Language (SQL/JRT)

TECHNICAL CORRIGENDUM 1

Foreword

1. *Rationale: Remove incorrect reference to obsolete part.*

In the 7th paragraph, delete the 5th bullet.

4 Concepts

4.3 Java class name resolution

1. *Rationale: Describe a JAR's SQL-Java path and its relationship to a JAR manifest's Class-Path attributes.*

Replace the 2nd paragraph with:

When JARs appear in the CLASSPATH, an ability exists for further effective extension of that CLASSPATH. Additional JARs will be included in the class resolution process when a JAR in the CLASSPATH has a manifest specifying one or more Class-Path attributes. A Class-Path attribute provides *relative URLs* of additional JARs. These Class-Path attribute URLs are *relative* to the source, for example the directory containing the JAR whose manifest is then being processed. A full URL, for example a `file:/` or `http://` format URL, is not allowed in a Class-Path attribute. The JARs enumerated by Class-Path attributes extend the CLASSPATH.

When a JVM is transitioned to being effectively within an SQL-environment, the problem of managing the JVM's class name resolution continues to exist, but with a change in emphasis. One important change is that an installed JAR manifest's Class-Path attributes cannot be honored. No relative URL has meaning when the source of the current JAR is given by a <catalog name>, <unqualified schema name>, and <jar id>. To allow the creators of Java applications a greater degree of control over class name resolution, and the added security associated with that control, a Class-Path attribute-like mechanism is defined to be a property of the JARs containing the Java applications, rather than as an environmental variable of the current session (such as, for example, CURRENT_PATH for dynamic statements). This mechanism, referred to as a JAR's SQL-Java path, provides a means for owners of installed JARs to control the class resolution process that the CLASSPATH and Class-Path attributes give users and creators of JARs outside an SQL-environment. But, note that these two mechanisms are only similar, they are not identical. If, while an external Java routine is being executed, a previously unseen class reference is encountered, that class is searched for in the JAR containing the definition of the currently executing class, and, if it is not found, the class will be sought in the manner specified by the SQL-Java path associated with that JAR (if any).