



**INTERNATIONAL STANDARD ISO/IEC 13249-5:2001**  
**TECHNICAL CORRIGENDUM 1**

Published 2003-08-15

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 multimedia and application packages —**

## **Part 5: Still Image**

### **TECHNICAL CORRIGENDUM 1**

*Technologies de l'information — Langages de bases de données — Multimédia SQL et paquetages d'application —*

*Partie 5: Image fixe*

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 13249-5:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

#### **Statement of purpose of rationale:**

A statement indicating the rationale for each change to ISO/IEC 13249-5:2001(E) is included. This is to inform the users of that standard as to the reason 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 Corrigendum introduces new Definitional Rules and Descriptions, 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 Corrigendum introduces new Subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, Subclause 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 13249-5:2001/Cor.1:2003(E)**

## Contents

Page

|                                                       |    |
|-------------------------------------------------------|----|
| Global changes .....                                  | 3  |
| 3.1.2 Definitions provided in Part 5 .....            | 3  |
| 4.1 Introduction .....                                | 3  |
| 4.7 The Still Image Information Schema .....          | 4  |
| 5.1.2 SI_StillImage Methods .....                     | 4  |
| 5.1.5 SI_Thumbnail Methods .....                      | 4  |
| 5.1.12 Functions not intended for Public Use .....    | 5  |
| 6.2.3 SI_Append Method .....                          | 5  |
| 6.3.1 SI_PositionalColor Type .....                   | 6  |
| 6.3.2 SI_PositionalColor Method .....                 | 7  |
| 6.3.3 SI_Score Method .....                           | 7  |
| 6.5.2 SI_FeatureList Method .....                     | 8  |
| 6.5.3 SI_setFeature Methods .....                     | 9  |
| 6.6.2 SI_RGBColor Method .....                        | 11 |
| 7.6 SI_VALUES view .....                              | 11 |
| 7.7 Short name views .....                            | 11 |
| 8.3 SI_IMAGE_FORMAT_CONVERSIONS base table .....      | 12 |
| 8.6 SI_VALUES base table .....                        | 12 |
| 9 Status Codes .....                                  | 13 |
| 10.1 Requirements for conformance .....               | 14 |
| Annex A.1 Implementation-defined Meta-variables ..... | 14 |
| Annex B Implementation-dependent elements .....       | 14 |

## Global changes

1. *Rationale: Align function names to use consistent naming conventions.*

Globally replace all the functions names listed as "Old Function Names" in the following table with corresponding function names listed as "New Functions Name".

| Old Function Name  | New Function Name  |
|--------------------|--------------------|
| SI_getThumbnail    | SI_getThmbnl       |
| SI_fndAverageColor | SI_findAvgClr      |
| SI_mkAverageColor  | SI_mkAvgClr        |
| SI_ScoreByAvgClr   | SI_ScoreByAvgClr   |
| SI_findColorHstgrm | SI_findClrHstgr    |
| SI_mkColorHistogr  | SI_mkClrHstgr      |
| SI_arrayClrHstgrm  | SI_arrayClrHstgr   |
| SI_appendClrHstgrm | SI_appendClrHstgr  |
| SI_findPositColor  | SI_findPstnlClr    |
| SI_ScoreByPositClr | SI_ScoreByPstnlClr |
| SI_mkFeatureList   | SI_mkFtrList       |
| SI_setAvgClrFtrW   | SI_setAvgClrFtr    |
| SI_setClrHstgrFtrW | SI_setClrHstgrFtr  |
| SI_setPstnlClrFtrW | SI_setPstnlClrFtr  |
| SI_setTextureFtrW  | SI_setTextureFtr   |
| SI_mkRGBColor      | SI_mkRGBClr        |

### 3.1.2 Definitions provided in Part 5

1. *Rational: The term "most significant color" was missing in the list of defined terms.*

Add the following definition:

#### 3.1.2.7a

##### **most significant color**

a color with the highest frequency in a color histogram

## 4.1 Introduction

1. *Rationale: Remove requirement for supported formats to be able to derive features from images with that format.*

Replace the sentence directly before NOTE 4 with:

An image format is a format supported by an implementation (for short: a supported format) if the implementation is able to derive the inherent image characteristics from the raw image.

## 4.7 The Still Image Information Schema

### 1. Rationale: Extend SI\_VALUES view.

Replace the fifth item, which describes the view SI\_VALUES, in the item list with:

- a view SI\_VALUES that lists implementation-defined meta-variable and their values.

## 5.1.2 SI\_StillImage Methods

### 1. Rationale: Correct the description of the condition when the exception condition is raised.

In the Description section, replace Description 3) b) with:

- b) Both of the following are False:
  - i) *explicitFormat* indicates a supported image format, and *explicitFormat* is equivalent to the format derived from *content*.
  - ii) *explicitFormat* indicates an unsupported image format, and the image format derived from *content* is the null value (i.e. no supported image format can be derived from *content*).

## 5.1.5 SI\_Thumbnail Methods

### 1. Rationale: Correct syntax errors in the Definition.

In the Definition, replace the definition for the method *SI\_Thumbnail*(INTEGER, INTEGER) with:

```
CREATE METHOD SI_Thumbnail
  (height INTEGER,
   width INTEGER)
RETURNS SI_StillImage
FOR SI_StillImage
BEGIN
  DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF12';

  IF height > SELF.SI_height OR
     width > SELF.SI_width OR
     NOT SI_supportedThumbnail(SELF.SI_format) = 1 THEN
    SIGNAL InvalidInput
      SET MESSAGE_TEXT =
        'illegal specification for thumbnail generation';
  END IF;
  RETURN SI_deriveThumbnail(SELF, height, width);
END
```

### 5.1.12 Functions not intended for Public Use

#### 1. Rationale: Correct a syntax error in the Definition.

Replace the definition of the function *SI\_supportedThumbnail* with:

```
CREATE FUNCTION SI_supportedThumbnail
  (sourceFormat CHARACTER VARYING(SI_MaxFormatLength))
  RETURNS INTEGER
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  CALLED ON NULL INPUT
  RETURN
    CASE
      WHEN EXISTS (
        SELECT *
          FROM SI_INFORMTN_SCHEMA.SI_THUMBNAIL_FORMATS
         WHERE TRIM(BOTH ' ' FROM SI_FORMAT) =
              TRIM(BOTH ' ' FROM sourceFormat)
        ) THEN 1
      ELSE 0
    END
```

#### 2. Rationale: Correct minor errors in the Description section.

Replace Description 8) with:

- 8) The function *SI\_deriveThumbnail*(*SI\_StillImage*, *INTEGER*, *INTEGER*):
- This function takes the following input parameters:
    - an *SI\_StillImage* value *image*,
    - an *INTEGER* value *height*,
    - an *INTEGER* value *width*.
  - This function derives and returns a thumbnail of its parameter *image*. If a thumbnail cannot be derived from *image* then the result *ret* is the null value. Otherwise, *ret* returns a thumbnail with the specified *height* and *width*. The format of the thumbnail *ret.SI\_format* is the same as the format of the still image *image*.

#### 3. Rationale: Correct minor errors in the Description section.

Replace Description 9) with:

- 9) The function *SI\_supportedThumbnail*(*CHARACTER VARYING*):
- This function takes the following input parameter:
    - a *CHARACTER VARYING* value *sourceFormat*.
  - If the view *SI\_THUMBNAIL\_FORMATS* of the information schema *SI\_INFORMTN\_SCHEMA* contains a row whose *SI\_FORMAT* column value is equivalent to *sourceFormat*, then the result of this function is 1 (one); i.e. a thumbnail can be derived from an image whose format indication equals *sourceFormat*. Otherwise, the result is 0 (zero); i.e. a thumbnail cannot be derived from an image whose format indication equals *sourceFormat*.

### 6.2.3 SI\_Append Method

#### 1. Rationale: Correct parameter name in the Definition.

Replace the Definition with:

```
CREATE METHOD SI_Append
  (color SI_Color,
```

```

    frequency DOUBLE PRECISION)
RETURNS SI_ColorHistogram
FOR SI_ColorHistogram
BEGIN
    DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF03';

    IF CARDINALITY(SELF.SI_ColorsList) = SI_MaxHistogramLength OR
       color IS NULL OR
       frequency IS NULL OR
       frequency < 0.0 OR frequency > 100.0 THEN
        SIGNAL InvalidInput
        SET MESSAGE_TEXT =
            'incorrect color histogram feature specification';
    END IF;
    SET SELF.SI_ColorsList =
        SELF.SI_ColorsList || ARRAY[color];
    SET SELF.SI_FrequenciesList =
        SELF.SI_FrequenciesList || ARRAY[frequency];
    RETURN SELF;
END

```

### 6.3.1 SI\_PositionalColor Type

1. *Rationale: Use the most significant color instead of the average color for the positional color feature.*

Replace the Definition with:

```

CREATE TYPE SI_PositionalColor
AS (
    SI_ColorPositions SI_Color ARRAY[SI_NumberSections]
)
INSTANTIABLE
NOT FINAL

CONSTRUCTOR METHOD SI_PositionalColor
(sourceImage SI_StillImage)
RETURNS SI_PositionalColor
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
CALLED ON NULL INPUT,

METHOD SI_Score
(image SI_StillImage)
RETURNS DOUBLE PRECISION
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT

```

2. *Rationale: Use the most significant color instead of the average color for the positional color feature.*

Replace the Description 2) with:

- 2) The *SI\_PositionalColor* type represents lists of *SI\_Color* values using the attribute:
  - a) an *SI\_Color* ARRAY value *SI\_ColorPositions*.

3. *Rationale: Use the most significant color instead of the average color for the positional color feature.*

Replace the Description 3) with:

- 3) The attribute *SI\_ColorPositions* is not for public use. There are no GRANT statements granting EXECUTE privilege on the observer and mutator functions for the attribute *SI\_ColorPositions*.

### 6.3.2 SI\_PositionalColor Method

1. *Rationale: Use the most significant color instead of the average color for the positional color feature.*

Replace the Description 3) with:

- 3) The method *SI\_PositionalColor(SI\_StillImage)* derives an *SI\_PositionalColor* value from the parameter *sourceImage*. To that end, *sourceImage* is effectively divided into *n* by *m* rectangles, and for each rectangle, the most significant color value is determined. The array, thus computed, of color values which represent most significant colors is the *SI\_ColorPositions* value of the returned *SI\_PositionalColor* value. Further details on the relationship between *sourceImage* and the resulting *SI\_PositionalColor* value, such as the values *n* and *m*, are implementation-dependent.

NOTE 1 The color histograms, from which the color values representing the most significant color for each rectangle are derived, are determined as described in Description 3) in Subclause 7, 6.2.2, "SI\_ColorHistogram Methods", for the method *SI\_ColorHistogram(SI\_StillImage)*.

### 6.3.3 SI\_Score Method

1. *Rationale: Use the most significant color instead of the average color for the positional color feature.*

Replace the Description 2) with:

- 2) The method *SI\_Score(SI\_StillImage)* returns a value greater than or equal to 0 (zero). For scoring an image, that image is effectively divided into *n* by *m* rectangles, such that the product of *n* and *m* equals *SI\_NumberSections*. The lower the returned value, the better the *n* by *m* most significant colors of *image* are characterized by the most significant colors represented by the *SI\_PositionalColor* value used for scoring *image*.

NOTE 2 The way in which *image* is divided into *SI\_NumberSections* of rectangles is implementation-dependent, as well *n* and *m* itself. However, the division shall be performed in the same fashion for the *SI\_Score* method and the method *SI\_PositionalColor(SI\_StillImage)*.

Case:

- a) If SELF or *image* or *image.SI\_content* is the null value, or if the positional color feature is not supported for *image*, then the null value is returned.
- b) Otherwise, the exact relationship between the values of *SI\_PositionalColor*, *SI\_StillImage* and the result of *SI\_Score(SI\_StillImage)* is implementation-dependent.

## 6.5.2 SI\_FeatureList Method

1. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Definition with:

```
CREATE CONSTRUCTOR METHOD SI_FeatureList
(averageColorFeature SI_AverageColor,
averageColorFeatureWeight DOUBLE PRECISION,
colorHistogramFeature SI_ColorHistogram,
colorHistogramFeatureWeight DOUBLE PRECISION,
positionalColorFeature SI_PositionalColor,
positionalColorFeatureWeight DOUBLE PRECISION,
textureFeature SI_Texture,
textureFeatureWeight DOUBLE PRECISION)
RETURNS SI_FeatureList
FOR SI_FeatureList
BEGIN
    DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF04';

    IF averageColorFeature IS NOT NULL AND
        (averageColorFeatureWeight IS NULL OR
         averageColorFeatureWeight < 0.0) OR
        colorHistogramFeature IS NOT NULL AND
        (colorHistogramFeatureWeight IS NULL OR
         colorHistogramFeatureWeight < 0.0) OR
        positionalColorFeature IS NOT NULL AND
        (positionalColorFeatureWeight IS NULL OR
         positionalColorFeatureWeight < 0.0) OR
        textureFeature IS NOT NULL AND
        (textureFeatureWeight IS NULL OR
         textureFeatureWeight < 0.0) THEN
        SIGNAL InvalidInput
        SET MESSAGE_TEXT = 'incorrect feature list specification';
    END IF;
    RETURN SELF.
    SI_AvgClrFtr(averageColorFeature).
    SI_AvgClrFtrWght(
        CASE
            WHEN averageColorFeature IS NULL THEN 0.0
            ELSE averageColorFeatureWeight
        END).
    SI_ClrHstgrFtr(colorHistogramFeature).
    SI_ClrHstgrFtrWght(
        CASE
            WHEN colorHistogramFeature IS NULL THEN 0.0
            ELSE colorHistogramFeatureWeight
        END).
    SI_PstnlClrFtr(positionalColorFeature).
    SI_PstnlClrFtrWght(
        CASE
            WHEN positionalColorFeature IS NULL THEN 0.0
            ELSE positionalColorFeatureWeight
        END).
    SI_TextureFtr(textureFeature).
    SI_TextureFtrWght(
        CASE
            WHEN textureFeature IS NULL THEN 0.0
            ELSE textureFeatureWeight
        END);
END
```

2. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Description 2) with:

- 2) If any of the parameters *averageColorFeature*, *colorHistogramFeature*, *positionalColorFeature*, or *textureFeature* is not the null value, and if any of the corresponding parameters *averageColorFeatureWeight*, *colorHistogramFeatureWeight*, *positionalColorFeatureWeight* or *textureFeatureWeight* is the null value, or less than 0 (zero), then an exception condition is raised: *SQL/MM Still Image exception – incorrect feature list specification*.

### 6.5.3 SI\_setFeature Methods

1. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Definition with:

```
CREATE METHOD SI_setFeature
(averageColorFeature SI_AverageColor,
 averageColorFeatureWeight DOUBLE PRECISION)
RETURNS SI_FeatureList
FOR SI_FeatureList
BEGIN
    DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF04';

    IF averageColorFeature IS NOT NULL AND
       (averageColorFeatureWeight IS NULL OR
        averageColorFeatureWeight < 0.0) THEN
        SIGNAL InvalidInput
        SET MESSAGE_TEXT = 'incorrect feature list specification';
    END IF;
    RETURN SELF.
    SI_AvgClrFtr(averageColorFeature).
    SI_AvgClrFtrWght(
        CASE
            WHEN averageColorFeature IS NULL THEN 0.0
            ELSE averageColorFeatureWeight
        END);
END

CREATE METHOD SI_setFeature
(colorHistogramFeature SI_ColorHistogram,
 colorHistogramFeatureWeight DOUBLE PRECISION)
RETURNS SI_FeatureList
FOR SI_FeatureList
BEGIN
    DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF04';

    IF colorHistogramFeature IS NOT NULL AND
       (colorHistogramFeatureWeight IS NULL OR
        colorHistogramFeatureWeight < 0.0) THEN
        SIGNAL InvalidInput
        SET MESSAGE_TEXT = 'incorrect feature list specification';
    END IF;
    RETURN SELF.
    SI_ClrHstgrFtr(colorHistogramFeature).
    SI_ClrHstgrFtrWght(
        CASE
            WHEN colorHistogramFeature IS NULL THEN 0.0
            ELSE colorHistogramFeatureWeight
        END);
END
```

```

CREATE METHOD SI_setFeature
  (positionalColorFeature SI_PositionalColor,
   positionalColorFeatureWeight DOUBLE PRECISION)
RETURNS SI_FeatureList
FOR SI_FeatureList
BEGIN
  DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF04';

  IF positionalColorFeature IS NOT NULL AND
     (positionalColorFeatureWeight IS NULL OR
      positionalColorFeatureWeight < 0.0) THEN
    SIGNAL InvalidInput
      SET MESSAGE_TEXT = 'incorrect feature list specification';
  END IF;
  RETURN SELF.
  SI_ClrHstgrFtr(positionalColorFeature).
  SI_ClrHstgrFtrWght(
    CASE
      WHEN positionalColorFeature IS NULL THEN 0.0
      ELSE positionalColorFeatureWeight
    END);
END

CREATE METHOD SI_setFeature
  (textureFeature SI_Texture,
   textureFeatureWeight DOUBLE PRECISION)
RETURNS SI_FeatureList
FOR SI_FeatureList
BEGIN
  DECLARE InvalidInput CONDITION FOR SQLSTATE '2FF04';

  IF textureFeature IS NOT NULL AND
     (textureFeatureWeight IS NULL OR
      textureFeatureWeight < 0.0) THEN
    SIGNAL InvalidInput
      SET MESSAGE_TEXT = 'incorrect feature list specification';
  END IF;
  RETURN SELF.
  SI_ClrHstgrFtr(textureFeature).
  SI_ClrHstgrFtrWght(
    CASE
      WHEN textureFeature IS NULL THEN 0.0
      ELSE textureFeatureWeight
    END);
END

```

2. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Description 2) with:

- 2) If the parameter *averageColorFeature* is not the null value, and if the parameter *averageColorFeatureWeight* is the null value or less than 0 (zero), then an exception condition is raised: *SQL/MM Still Image exception – incorrect feature list specification*.

3. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Description 5) with:

- 5) If the parameter *colorHistogramFeature* is not the null value, and if the parameter *colorHistogramFeatureWeight* is the null value or less than 0 (zero), then an exception condition is raised: *SQL/MM Still Image exception – incorrect feature list specification*.

4. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Description 8) with:

- 8) If the parameter *positionalColorFeature* is not the null value, and if the parameter *positionalColorFeatureWeight* is the null value or is less than 0 (zero), then an exception condition is raised: *SQL/MM Still Image exception – incorrect feature list specification.*

5. *Rationale: Allow null values for the weight attributes if the respective feature values are null values.*

Replace the Description 11) with:

- 11) If the parameter *textureFeature* is not the null value, and if the parameter *textureFeatureWeight* is the null value or less than 0 (zero), then an exception condition is raised: *SQL/MM Still Image exception – incorrect feature list specification.*

## 6.6.2 SI\_RGBColor Method

1. *Rationale: Correct wording in the Description section.*

Replace Description 2) with:

- 2) If any of the parameters is the null value, or if any of the input values are less than 0 (zero) or greater than *SI\_MaxRGBColor*, then an exception condition is raised: *SQL/MM Still Image exception – incorrect color specification.*

2. *Rationale: Correct wording in the Description section.*

Replace Description 3) with:

- 3) It is implementation-dependent how the color value in the RGB color space, specified by its red, green, and blue components, is represented by the implementation-dependent set of attributes of the *SI\_Color* type.

## 7.6 SI\_VALUES view

1. *Rationale: Extend SI\_VALUES view.*

Replace the body of this Subclause with:

### Purpose

List the implementation-defined meta-variable and their values.

### Definition

```
CREATE VIEW SI_VALUES AS
SELECT SI_VALUE, SI_SUPPORTED_VALUE
FROM SI_DEFINITION_SCHEMA.SI_VALUES
```

## 7.7 Short name views

1. *Rationale: Correct column names in the definition of the view SI\_IMAGE\_FRMTS\_FTRS.*

Replace the definition of the view *SI\_IMAGE\_FRMT\_FTRS* with:

```
CREATE VIEW SI_IMAGE_FRMT_FTRS AS
SELECT SI_FORMAT, SI_FEATURE_NAME
FROM SI_DEFINITION_SCHEMA.SI_IMAGE_FORMAT_FEATURES
```