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)**

Published in Switzerland

Contents

Page

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_ScoreByAvrgClr	SI_ScoreByAvgClr
SI_findColorHstgrm	SI_findClrHstgr
SI_mkColorHistogrm	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_height 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)*:

   a) This function takes the following input parameters:

      i) an *SI_StillImage* value *image*,

      ii) an INTEGER value *height*,

      iii) an INTEGER value *width*.

   b) 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)*:

   a) This function takes the following input parameter:

      i) a CHARACTER VARYING value *sourceFormat*.

   b) 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
```