

UK National Body Comments on
Spatial Reference Model (SRM) Language Binding to C
Final Committee Draft ISO/IEC 18042-4
(ISO/IEC JTC 1/SC 24 N2594)
(ISO/IEC JTC 1/SC 24 WG 8 N0366)

The UK votes to DISAPPROVE CD 18042-4 for the reasons given below. Acceptance of these reasons and appropriate changes in the text will change the vote to APPROVAL.

General

The following comments apply to the entire document:

UK_G001:

Entire document

The content of this document should track the changes in ISO/IEC 18026 and should progress prior to the progression of ISO/IEC 18026.

UK_G002:

Entire document

Per the most recent guidance from the ISO editors, all content except for the “image.html” file should be placed in a separate lower subdirectory.

UK_G003:

Entire document

Per the most recent guidance from the ISO editors, all hyphens should be removed from file names. It is suggested that the phrases “Clause x --” and “Annex x --” should be removed from the files names. All cross-file hyperlinks should be updated as necessary.

Technical

Clause 6—Object class definitions

UK_T001:

6.3.9, SRM_EC_Method_List

The type SRM_CreateLocalTangentPlaneEuclideanSRF should be SRM_CreateLocalTangentSpaceEuclideanSRF and the field name

CreateLocalTangentPlaneEuclideanSRF should be CreateLocalTangentSpaceEuclideanSRF.
This change should be applied throughout the document.

Annex A—Bound constructs in compilation order

UK_T002:

A.3

The SRM_Position2D and SRM_Position3D classes should be moved to immediately follow SRM_LifeCycleObject so that they may be referenced correctly in subsequent definitions.

UK_T003:

A.3

The LTSE class should be moved to immediately precede SRM_BaseSRFWithEllipsoidalHeight so that the reference to SRM_LocalTangentSpaceEuclidean can be satisfied.

Editorial

Clause 2—Normative references

UK_E001:

Table

The table should be centered.

UK_E002:

Table

The hyperlinks should be attached to the text “ISO/IEC” in the document number for all entries in the “Reference” column.

UK_E003:

Table

The names of the standards should be italicized.

Clause 4—Tables

UK_E004:

4.3.3, Table 4.4

In column 3, “namesnames” should be “names”.

UK_E005:

4.3.3, Table

This is the second occurrence of Table 4.4. This table should be renumbered as Table 4.5.

Clause 5—Type definitions

UK_E006:

5.2.2

The code for the type SRM_Object_Reference is too small.

Clause 6—Object class definitions

UK_E007:

6.3, throughout

The phrase “Abstract Function” should be capitalized “Abstract function”.

Annex A—Bound constructs in compilation order

UK_E008:

A.3

The formatting of the text following SRM_CreateDirection is in too small a font. This is probably due to not closing a <pre> construct in SRM_CreateDirection

JAPANESE COMMENTS

Date:

2004-10-20

Comments:

The national body of Japan disapproves FCD 18024-4 (SRM C binding) for reason as below. Acceptance of these reasons and appropriate changes in the text will change our vote to approval.

JAPAN_T001) This document should be changed according to the changes in FCD 18026.

**SEDRIS Organization Comments on
Spatial Reference Model
Language Binding
ISO/IEC FCD 18042-4
Submitted: 11 November 2004**

GENERAL

SEDRIS_G001:

Throughout

There should be no logos, other than ISO and IEC. Remove the SRM logo at the top of the index.html page.

SEDRIS_G002:

Tables 3.1 and 4.1

Tables 3.1 and 4.1 are titled "Topics", whereas Tables 5.1, 6.1, A.1 and B.1 are titled "Table of contents". All these table titles should be consistently worded.

SEDRIS_G003:

index.html, Annex A, Annex B

In checking the W3C html validation (at www.w3.org), these files failed the validation check and yielded the following error message: **This page is not Valid [HTML 4.01 Transitional](#)!** These pages need to be updated to conform correctly.

TECHNICAL

Clause 3

SEDRIS_T001:

3

No reference is made to [I10641].

Remove it.

SEDRIS_T002:

3.2 Conformance

The conformance clause states that “In order to conform, an implementation of the C binding of the SRM API shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 18042”, but there are optional abstract classes and private methods defined in the SRM abstract spec that should not be required for implementation.

Add clarification to the text to those exceptions.

SEDRIS_T003:

3.4 Character strings

This clause seems to be describing a property of the C programming language and not specifically related to any aspect of the SRM.

Remove this clause.

SEDRIS_T004:

3.7 Returned values

All SRM API functions return SRM_StatusCode without exceptions.

Change the word “most” to “all” in the first paragraph and remove the second paragraph.

SEDRIS_T005:

3.8 Header files – 1st paragraph

The requirement for the inclusion order for the srm_types.h and srm_object.h files is mute. The srm_types.h is already included in srm_object.h.

Remove that requirement.

SEDRIS_T006:

3.8 Header files – 2nd paragraph

There is no description of the srm_functions.h file. It was probably meant srm_objects.h

Replace “srm_functions.h” with “srm_objects.h”.

SEDRIS_T007:

3.8 Header files – 3rd paragraph

If the comment in Annex A (*4) to remove the SRM_SRF_Info structure from the spec is accepted, then dependency on edcs.h is removed.

Remove this paragraph if the comment in Annex A (*4) is accepted.

SEDRIS_T008:

Table 3.2 – C-specific SRM errors

Table 3.2 does not align with the status code defined in SRM abstract spec. Clause 11.

Change Table 3.2 with the items from SRM abstract spec status code in clause 11.2.5.2.

SEDRIS_T009:

3.9.1 Error codes

The sentence:

"All errors reported by an implementation are returned by functions using the data type SRM_StatusCode" is ambiguous.

Does it mean errors are reported "using the data type SRM_StatusCode"?

Or does it mean reported errors are only returned by those functions that

use "the data type SRM_StatusCode"?

Change the statement to say: "All function implementation return error status of data type SRM_Status_Code".

SEDRIS_T010:

3 Footnote

"...the ISO council has designated the NGA to act as registration authority"

In the Florida meeting, NGA preferred to be referred to as "NIMA" for this

purpose.

WG8 should confirm this with NGA and the same name should be used for both 18026 and 18042-4.

Clause 4

SEDRIS_T011:

4.2.1 Abbreviation policy – item c

The text refers to "refinements" (in plural) while only one was specified. It is not clear what was meant.

Clarify the text, perhaps with an example.

SEDRIS_T012: (*3)

Table 4.2 – Table of abbreviations

The following abbreviation entries are not used: direction of forward, direction of up.

All the abbreviations related to HSR are to exclude the acronym “HSR” as part of abbreviation if the comment to put all HSR codes into a single code space is adopted. The reason being, the HSR would be the type name for all the HSR enumerations and not part of the HSR enumeration-specific names.

The abbreviation for “ObliqueMercatorSpherical” should have been “OMS”.

The abbreviation for “SolarEcliptic” should have been “SEC”.

The abbreviation for “StatusCode” should have been “STATCOD”.

Remove the entries for direction of forward and direction of up.

Remove “HSR” from the HSR enumeration abbreviations.

Change abbreviation for “ObliqueMercatorSpherical” from “OM” to “OMS”.

Change abbreviation for “SolarEcliptic” from “SE” to “SEC”.

Change abbreviation for “StatusCode” from “STSC” to “STATCOD”.

SEDRIS_T013:

Table 4.3, 4.4 – Data type names

The HSR_Code binding should change from one-to-many to one-to-one if the 18026 comment to put all such codes into a single code space is adopted.

SEDRIS_T014:

Table 4.3, 4.4 – Data type names

The heading "Abstract name" can be confused with the concepts of "abstract object" and "concrete object".

The "Abstract name" should be changed to "Abstract SRM API name".

SEDRIS_T015:

Table 4.4 – Object class names

Wrong table number.

Missing method “CreateDirection” in BaseSRF3D class.

Wrong method name “EuclideanDistanceSurface” in BaseSRFwithTangentPlaneSurface.

Missing “VerticalOffset” method from BaseSRFwithEllipsoidalHeight class.

Wrong use of abbreviation “OM” in bound names for ObliqueMercatorSpherical.

Wrong use of abbreviation “SE” in bound names for SolarEcliptic.

Change table number from “4.4” to 4.5”.

Add method “CreateDirection” in BaseSRF3D class.

Change method name “EuclideanDistanceSurface” to “EuclideanDistance” in BaseSRFwithTangentPlaneSurface.

Add method “VerticalOffset” to BaseSRFwithEllipsoidalHeight class.

Change occurrences of abbreviation “OM” to “OMS” in the ObliqueMercatorSpherical entry.

Change occurrences of abbreviation “SE” to “SEC” in the SolarEcliptic entry.

Clause 5

SEDRIS_T016: (*1)

5.2.4 Selection data types

Wrong abbreviation used for StatusCode in SRM_Status_Code enumeration.

The status code of “BUFFER_OVERFLOW” and “ENUMERATION_VALUE_INVALID” is equivalent to “FLOATINGPOINT_ERROR” and “MEMORY_ALLOCATION_ERROR” respectively, and the former ones do not exist in the abstract spec.

Change all occurrences of “STSC” to “STATCOD”

Remove enumeration “BUFFER_OVERFLOW” and “ENUMERATION_VALUE_INVALID” from the SRM_Status_Code.

SEDRIS_T017: (*2)

5.2.5 Integer code types

If the 18026 comment is accepted to have all the HSR_Code defined within a single code space, then there is only one HSR_Code data type that needs to be defined.

The SRF_Code enumerations do not align with the abstract spec.

Remove all the different HSR_Code related data types and replace them with once single SRM_HSR_Code type that contains all the HSR enumerations.

Change “SRM_SRFCOD_GEODETTIC_EUROPE_1950_7” to “SRM_SRFCOD_GEODETTIC_N_AMERICAN_1984”.

Change “SRM_SRFCOD_GEODETTIC_N_AMERICAN_1983_8” to “SRM_SRFCOD_GEODETTIC_WGS_1984”.

Clause 6

SEDRIS_T018:

Throughout

The following parameter in all the SRF creation methods should be changed from
'orm_code SRM_ORM' to
'orm_code SRM_ORM_Code'.

SEDRIS_T019:

Throughout

Missing hsr_code parameter from all the SRF creation methods requiring the orm_code parameter.

Add the following parameter after the orm_code parameter in all the SRF creation methods except for the SRM_CreateStandardSRF:

'hsr_code SRM_HSR_Code'.

SEDRIS_T020:

Throughout

All SRF creation methods should not be 'typedef', but actual function declarations. Remove the reserved word 'typedef' from declaration of 'SRM_XX_Create', SRM_CreateSRFSetMember and SRM_CreateStandardSRF methods.

SEDRIS_T021:

Throughout

SRM_Object_Reference is already a 'void *', yet all parameters of type SRM_Object_Reference are used as if they were non-pointer types. Change all SRM_Object_Reference parameters so that incoming parameters have no '*' and outgoing parameters have only one '*'.

SEDRIS_T022:

Throughout

Using 'this' as parameters to methods should be avoided. Change 'this' to 'this_object'.

SEDRIS_T023:

Throughout

SRM_GetCSCCode defined in section 6.2.3, yet 'GetCSCodes', plural, is used. Since the method defined in section 6.2.3 returns a single code, the rest of the references to this method should be singular. Change all references to 'GetCSCodes' to 'GetCSCCode'.

SEDRIS_T024:

Throughout

SRM_Coordinate types are not consistent throughout in regards to the pointers. Change all SRM_Coordinate parameters so that they all have one '*'.

SEDRIS_T025:

Throughout

SRM_Direction is not consistent throughout in regards to the pointers. Change all SRM_Direction parameters so that they all have one '*'.

SEDRIS_T026:

Throughout

Incorrect listing of 'LocalTangentSpaceEuclidean'. Change all occurrences of 'LocalTangentPlaneEuclidean' with 'LocalTangentSpaceEuclidean'.

SEDRIS_T027:

Throughout

Incorrect label 'EquidistanceCylindrical'.

Change all occurrences of 'EquidistanceCylindrical' to 'EquidistantCylindrical'.

SEDRIS_T028:

Throughout

Method lists for SRM_Coordinate and SRM_Direction types have lower case 'destroy' method. Change all occurrences of 'destroy' in SRM_Coordinate types and SRM_Direction to 'Destroy' for consistency.

SEDRIS_T029:

6.2 and 6.3 throughout

The terms "Abstract Function" and "Abstract object class" can be confused with the concepts of "abstract object" and "concrete object".

The "Abstract Function" should be changed to "Abstract SRM API function" and "Abstract object class" should be changed to "Abstract SRM API object class".

SEDRIS_T030:

6.2.3 BaseSRF

In SRM_GetSRFCodes, the parameter 'SRM_SRF_Codes_Info *srf_codes_info' does not exist. Change 'SRM_SRF_Codes_Info *srf_codes_info /* OUT srf codes info */' to

```
'SRM_SRFT_Code          *srf_template_code,  
SRM_SRF_Code           *srf_code,  
SRM_SRFS_Code          *srf_set_code,  
SRM_SRFS_Member_Code  *srf_set_member_code'
```

SEDRIS_T031:

6.2.7 BaseSRFwithEllipsoidalHeight

Extra '*' on parameter 'local_tangent_Euclidean_srf' in method SRM_CreateLocalTangentSpaceEuclideanSRF.

Remove one '*' on parameter 'local_tangent_Euclidean_srf' in method SRM_CreateLocalTangentSpaceEuclideanSRF.

SEDRIS_T032:

6.3.13 LambertConformalConic

SRM_LCC_Parameters parameter is missing from the SRM_LCC_Create method.

Add the following as the second parameter to the SRM_LCC_Create method 'const SRM_LCC_Parameters parameters,'.

Add the following as the second parameter to the SRM_LTSC_Create method 'const SRM_LT_Parameters parameters,'.

SEDRIS_T039:

6.3.19 LocalTangentSpaceCylindrical

'SRM_LT_Parameters **parameters' has an extra '*' in SRM_LTSC_GetSRFParameters method.

Change 'SRM_LT_Parameters **parameters' to 'SRM_LT_Parameters *parameters'.

SEDRIS_T040:

6.3.20 LocalTangentSpaceEuclidean

SRM_LTSE_Parameters parameter is missing from the SRM_LTSE_Create method.

Add the following as the second parameter to the SRM_LTSE_Create method 'const SRM_LTSE_Parameters parameters,'.

SEDRIS_T041:

6.3.20 LocalTangentSpaceEuclidean

'SRM_LTSE_Parameters **parameters' has an extra '*' in SRM_LTSE_GetSRFParameters method.

Change 'SRM_LTSE_Parameters **parameters' to 'SRM_LTSE_Parameters *parameters'.

SEDRIS_T042:

6.3.21 Mercator

SRM_Mercator_Parameters parameter is missing from the SRM_M_Create method.

Add the following as the second parameter to the SRM_M_Create method 'const SRM_Mercator_Parameters parameters,'.

SEDRIS_T043:

6.3.22 ObliqueMercatorSpherical

SRM_OM_Parameters parameter is missing from the SRM_OM_Create method.

Add the following as the second parameter to the SRM_OM_Create method 'const SRM_OM_Parameters parameters,'.

SEDRIS_T044:

6.3.32 TransverseMercator

SRM_Mercator_Parameters parameter is missing from the SRM_TM_Create method.

Add the following as the second parameter to the SRM_TM_Create method 'const SRM_Mercator_Parameters parameters,'.

Annex A

SEDRIS_T045:

A.

```
typedef SRM_Integer SRM_Status_Code
```

Make similar changes to the ones in Clause 4 above. (*1)

SEDRIS_T046:

A.

If the 18026 comment is accepted to have all the HSR_Code defined within a single code space, then there is only one HSR_Code data type that needs to be defined.

Remove all the specific SRM_HSR_XX types and replace them with a single SRM_HSR_Code type containing all the HSR enumerations as done in Clause 4 (*2 and *3)

SEDRIS_T047:

A.

The specification of SRM_ORM_Transformation_2D_Parameters structure is not consistent with the abstract spec.

Delete parameters 'omega_2' and 'omega_3'.

SEDRIS_T048: (*4)

A.2

Bound constructs in srm_type.h

The SRM_SRF_Info structure is not necessary for the definition of SRM information. That structure is implementation dependent and should be moved from the srm_types.h and into SEDRIS part 1.

Delete the SRM_SRF_Info structure. Also remove the SRM_Info data structure from the abstract spec. clause 11.

Annex B

SEDRIS_T049:

B.

Add the following example code corresponding to the examples in the SRM abstract spec:

B.4 Computing Euclidean distance:

```
#include "srm.h"
```

```
SRM_Status_Code status = SRM_STAT_CODE_SUCCESS;
```

```
SRM_Celestiodetic ex1_srf;
```

```
SRM_Coordinate3D coordinate1, coordinate2;
```

```
SRM_Long_Float distance = 0.0;
```

```
status = SRM_CD_Create(SRM_ORM_N_AM_1983,
```

```
SRM_HSR_N_AM_1983_CONTINENTAL_US,
```

```
&ex1_srf);
```

```
status = ex1_srf.methods->CreateCoordinate3D(&ex1_srf,  
                                             (-77 * (3.14 / 180)),  
                                             (38 * (3.14 / 180)),  
                                             0.0,  
                                             &coordinate1);
```

```
status = ex1_srf.methods->CreateCoordinate3D(&ex1_srf,  
                                             (3 * (3.14 / 180)),  
                                             (49 * (3.14 / 180)),  
                                             0.0,  
                                             &coordinate2);
```

```
status = ex1_srf.methods->EuclideanDistance3D(&ex1_srf,  
                                              &coordinate1,  
                                              &coordinate2,  
                                              &distance);
```

```
fprintf(stdout, "The Euclidean distance between coordinate1 and coordinate2 is %lf\n",  
distance);
```

```
status = coordinate1.methods->Destroy(&coordinate1);
```

```
status = coordinate1.methods->Destroy(&coordinate1);
```

```
status = ex1_srf.methods->Destroy(&ex1_srf);
```

B.5 Change a coordinate from a standard SRF to a SRF set member:

```
#include "srm.h"
```

```
SRM_Status_Code status = SRM_STAT_CODE_SUCCESS;
```

```
SRM_Coordinate3D source_coordinate, target_coordinate;
```

```
SRM_BaseSRF source_srf, target_srf;
```

```
SRM_Coordinate_Valid_Region region;
```

```
status =
```

```
    SRM_CreateSRFSetMember(SRM_SRFS_UNIVERSAL_TRANSVERSE_MERCATOR,  
                           SRM_SSM_UTM_ZONE_23_NORTHERN_HEMISPHERE,  
                           SRM_ORM_N_AM_1983,  
                           SRM_HSR_N_AM_1983_CONTINENTAL_US,  
                           &source_srf);
```

```
status =
```

```
    ((SRM_TransverseMercator *)&source_srf)->  
        methods->CreateCoordinate3D(&source_srf,  
                                     350000.0,  
                                     400.0,  
                                     0.0,  
                                     &source_coordinate);
```

```
status = SRM_CreateStandardSRF(SRM_SRF_GEOCENTRIC_WGS_1984,  
                               &target_srf);
```

```
status = ((SRM_Celestiocentric *)&target_srf)->
    methods->ChangeCoordinate3DSRF(&target_srf,
        &source_srf,
        source_coordinate,
        &target_coordinate,
        &region);
```

```
status = ((SRM_TransverseMercator *)&source_srf)->
    methods->DestroyCoordinate3D(&source_coordinate);
```

```
status = ((SRM_Celestiocentric *) &target_srf)->
    methods->DestroyCoordinate3D(&target_coordinate);
```

```
status = ((SRM_TransverseMercator *)&source_srf)->
    methods->Destroy(&source_srf);
```

```
status = ((SRM_Celestiocentric *) &target_srf)->
    methods->Destroy(&target_srf);
```

EDITORIAL

index.html file

SEDRIS_E001:

Title

In the title at the top of the page, change “ISO/IEC Committee Draft 18042-4” to read “ISO/IEC Final Committee Draft 18042-4” (insert the missing word “Final”).

SEDRIS_E002:

Hyperlink in index.html

The hyperlink to the ISO web site at the bottom of this file/page (and all other clauses) is to a 18023-2 URL, not to a 18042-4 URL. Correct the URL hyperlink (the URL text is correct).

SEDRIS_E003:

List item 1

In list item #1 following the Table of Contents, the statement “**Scope** defines the problem area that this International Standard addresses” does not agree with what the Directives, Part 2 says that a Scope statement does. It also does not agree with the contents of Scope because Scope identifies no “problem area”. The sentence needs to be reworded by the Editors to be correct. For reference, please review ISO Directives, Part 2, subclause 6.2.1 Scope.

Foreword

SEDRIS_E004:

Fourth Paragraph

In the 4th paragraph, change “teh” to read “the”, and change “irghts” to read “rights”.

SEDRIS_E005:

Last Paragraph

In the last paragraph, change “*Spatial Reference Model (SRM) language bindings*” to read “*Spatial reference model (SRM) language bindings*” (ISO capitalization).

Introduction

SEDRIS_E006:

Second paragraph

In the 2nd paragraph, provide a hyperlink to ISO/IEC 18026 in clause 2.

Clause 1

SEDRIS_E007:

First paragraph

In the 1st paragraph, provide a hyperlink to ISO/IEC 18026 in clause 2, change “language independent” to read “language-independent” (add missing hyphen), change “application programmer interface” to read “application program interface”, and change (2 places) “language dependent” to read “language-dependent” (add missing hyphen).

Clause 3

SEDRIS_E008:

3.2

In the 1st paragraph, change “application programmer interface” to read “application program interface”.

SEDRIS_E009:

Foonote

In the footnote, provide a hyperlink to the cited registration authority, NGA ([ISO International Registration Authority for Graphical Items](#) was used in the EDCS FDIS).

SEDRIS_E010:

3.8 Header files

The first sentence starts with a letter “C” which is not consistent with the prior references to the C programming language.

Change “C” to “The C language” in the first sentence.

Clause 4

SEDRIS_E011:

4.2.1

In the 1st paragraph, change “used in other [ISO C](#) binding standards” to read “as defined in [ISO/IEC 9899](#)”.

SEDRIS_E012:

4.2.1.a

Change “score” to read “underscore”.

SEDRIS_E013:

4.2.1.d, 4.2.1.e

Change “upper case” to read “upper-case” (3 places).

SEDRIS_E014:

4.3.3

Change the reference to [Table 4.4](#) to read [Table 4.5](#), and change the title of the table in this subclause to read “Table 4.5”.

SEDRIS_E015:

4.3.3, Table 4.4 (should be 4.5)

In the header row, 3rd column, change “namesnames” to read “names”.

Clause 5

SEDRIS_E016:

Table 5.1

In the 2nd column, change the hyperlink text (the URL is correct) from “[5.2.5 Structured data types](#)” to read “[5.2.6 Structured data types](#)”.

SEDRIS_E017:

5.1.2

Change “implementation dependent” to read “implementation-dependent”, and change “implementation independent” to read “implementation-independent” (insert missing hyphens in both places).

Clause 6

SEDRIS_E018:

Table 6.1

The [6.3.28 SolarEquatorial](#) hyperlink takes you to the SolarEquatorial function, rather than to the SolarEquatorial subclause. Correct the hyperlink.

SEDRIS_E019:

Table 6.1

The [6.3.30 SolarMagneticEcliptic](#) hyperlink takes you to the SolarMagneticEcliptic function, rather than to the SolarMagneticEcliptic subclause. Correct the hyperlink.

SEDRIS_E020:

6.1.2 Notational conventions

Spelling of 'parameter' in comments in SRM_CbindingMethodName example.

Change 'paramater' to 'parameter'.

SEDRIS_E021:

6.2.4 BaseSRF2D

Incorrect label on parameter in method SRM_GetCoordinate2DValues, 'new_coordinate'.

Change 'new_coordinate' parameter to 'coordinate' in method SRM_GetCoordinate2DValues.

SEDRIS_E022:

6.2.5 BaseSRF3D

Incorrect label on parameter in method SRM_SetExtendedValidRegion, 'coordinate'.

Change 'coordinate' parameter to 'component' in method SRM_SetExtendedValidRegion.

SEDRIS_E023:

6.2.5 BaseSRF3D

Incorrect label on parameter in method SRM_SetValidRegion, 'coordinate'.

Change 'coordinate' parameter to 'component' in method SRM_SetValidRegion.

SEDRIS_E024:

6.2.5 BaseSRF3D

Parameter list for method SRM_ChangeDirectionSRFObject has two parameters named 'source_direction'.

The second of the two 'source_direction's, of type SRM_ORM_Transformation_3D_Parameters, should be changed to 'h_st'.

SEDRIS_E025:

6.3.5 – 6.3.7, 6.3.25, 6.3.26, 6.3.31

There appears to be explanatory verbiage missing before the second typedef struct statements in each of these subclauses (e.g., "The following record type specifies the binding of the ...").

SEDRIS_E026:

6.3.12 HeliosphericEarthEquatorial

Incorrect name of 'SRM_HEEC_Method_List'.

Change 'SRM_HEEC_Method_List' to 'SRM_HEEQ_Method_List'.

SEDRIS_E027:

6.3.17 LocalSpaceRectangular3D

Incorrect name of 'SRM_LSP_Create'.

Change 'SRM_LSP_Create' to 'SRM_LSR3D_Create'.

SEDRIS_E028:

6.3.32 TransverseMercator

Wrong label of 'SRM_ObliqueMercator' in method SRM_TM_Create.

Change 'SRM_ObliqueMercator' to 'SRM_TransverseMercator' in method SRM_TM_Create.

SEDRIS_E029:

6.4.2, 6.4.3

Change the [6.3 SRM Concrete object classes](#) hyperlink text (the URL is correct) to read [6.3 SRM concrete object classes](#) (ISO capitalization).