WWNWS
Meeting 11
Agenda Item 6

S-124WS/2/6
12 August 2019

S-124 Data Quality and tests

Submitted by Canadian Coast Guard

SUMMARY

Executive Summary: [Insert summary]

Action to be taken: [Insert paragraph number in document where actions are requested]

Related documents: [Insert details of relevant documents or state "None"]

1. HSSC10/37 states that *HSSC WGs and PTs' Chairs to keep the DQWG informed on the DQWG recommendations related to Data Quality that have not been taken into account in the development of their Product Specifications [so the DQWG does not need to reiterate these recommendations at periodic reviews]*.

2. The draft of S-97 Part C (https://www.iho.int/mtg_docs/com_wg/S-100WG/S-100WG4/S100WG4_2019_6.2_EN_S-97_Edition_0.0.2_Guidebook_v1.zip) provides guidance for how develops product specification based on S-100 should account for data quality. The introduction says; "*When drafting a Product Specification, the Data Quality Checklist will serve as a guidance document to verify if the appropriate Data Quality Elements have been included in the Product Specification. A Data Quality Element is a quantitative component documenting the quality of a dataset. The applicability of a Data Quality Element to a dataset depends on both the dataset's content and its Product Specification, the result being that all available Data Quality Elements may not be applicable to all datasets"*. A challenge for S-124 datasets is that due to the urgency of the information, datasets may be based on incomplete or unconfirmed information and mariners will need to take this into account when deciding what reliance to place on the information contained therein. It is often not possible to determine quantifiable values to measures of data quality. In the draft S-124 product specification the following statement is included under data quality;

   > Generally the quality of information is evident from the navigational warning text by the use of qualitative words such as 'approximate', 'reported', 'in the vicinity of' and 'about'.

   Is it sufficient? Or should there be an attempt at adding something more quantifiable? If yes, how should it be done considering the sources are often limited and incomplete against the urgency of releasing the information?

3. Validation checks are recommended and the main purpose of them is to ensure the integrity of the dataset against the product specifications so to provide for smooth operation for the users. Draft version 2.0.0. of S-124 includes a preliminary set of validation checks adopted from other GML based product specifications. Experience with

similar checks in S-57 ENCs show that the number of tests increase over time and the tests themselves evolve with production and data use experience. It is therefore reasonable to assume the same will happen with S-124. Is the current list a sufficient start? Or should it be amended? Appendix E, which includes the checks and the geospatial definitions is included below.

Appendix E. Validation Checks

1. References

IHO S-58 ENC VALIDATION CHECKS Edition 6.1.0, September 2018
IHO S-97 Part C IHO data quality checklist [Draft 0.2, August 2018]

2. Abbreviation

PS – Product Specification
DCEG – Data Capture and Encoding Guide

3. Production validation checks for S-124 Navigational Warnings

The following checks are intended for production systems designed to produce S-124 Navigational Warning datasets. The checks can be administered at any time during the production phase. All checks should be considered as warnings, even though more severe classifications are available, due to the status of the development and lack of experience with system use of S-124 datasets, it is considered premature to classify any checks as error or critical error at this time. All operators and spatial expressions are defined in Annex A.

3.1 Check classification

C	Critical Error	An error which would make an MTM dataset unusable in ECDIS through not loading or causing an ECDIS to crash or presenting data which is unsafe for navigation.
E	Error	An error which may degrade the quality of the MTM dataset through appearance or usability but which will not pose a significant danger when used to support navigation.
W	Warning	An error which may be duplication or an inconsistency which will not noticeably degrade the usability of an MTM dataset in ECDIS.

3.2 Checks relating to S-124 Product Specification

No	Check description	Check message	Check solution	Conformity to
100	For each feature object where its geometry is not COVERED_BY the bounding box	Objects fall outside the coverage object.	Ensure objects are not outside of the limits of the cell.	GML schema
101	If the cell file size is greater than 50KB.	The cell is larger than 50KB in size.	Ensure that the cell is not larger than 50KB.	PS 11.7
102	For each feature record where the name is not unique WITHIN the dataset.	Duplicate FOIDs exist within the dataset.	Ensure that no duplicate FOIDs exist.	PS 10.8
103	If any mandatory attributes are not present.	Mandatory attributes are not encoded.	Populate mandatory attributes with a value.	PS 10.6

Commented [ME1]: Needs update to match the ID structure of NW

105	For each feature object with an attribute of type Float or Integer where the value contains zeroes before the first numerical digit or after the last numerical digit.	Values have been padded with non-significant zeroes. E.g. : For a flip fearing of 180 degrees, the value of flipBearing must be 180 and not 0180.00.	Remove non-significant zeroes.	PS 10.3
106	For each association between features instances, features instances and information instances, and between information instances that is not defined in the feature catalogue.	Wrong association used.	Use correct association type.	Logical consistency
107	For each role name on associations that is not defined in the feature catalogue.	Wrong role used.	Use correct role name.	Logical consistency
108	For each association that is not defined in the feature catalogue.	Unknown association is used.	Use association that is defined in the feature catalogue.	Logical consistency
109	For each role name that is not defined in the feature catalogue.	Unknown role name is used.	Use role name that is defined in the feature catalogue.	Logical consistency
110	For each association ensure associated classes are only those permitted by the feature catalogue.	Class is associated in an illegal association.	Ensure correct association is used between classes.	Logical consistency
111	For each role name ensure it is only used with permitted associations.	Role name is used on an illegal association.	Ensure correct role names are used on the association.	Logical consistency
112	Ensure dataset conformance to the GML schema.	Dataset does not conform to the GML schema.	Ensure conformance to the GML schema.	PS 10.1.1 & 11.1
113	Ensure all text fields are encoded using UTF-8.	Illegal character set used.	Change character encoding to UTF-8.	PS 10.4
114	For each feature instance where more than one featureName is present, and the name subattribute of two or more featureName instances is equal.	Values name sub attribute are identical.	Ensure that name subattributes are populated with the correct values.	Logical consistency
115	For each featureName subattribute with language not equal to eng, and where featureName subattributes with language equal to eng is not present.	Name is encoded in national language only.	Populate text attribute with English text.	Logical consistency
116	For each warning information subattribute with language not equal to eng, and where information subattribute with language equal to eng is not present.	Text is encoded in national language only	Populate name attribute with English text.	Logical consistency
117	If the horizontalDatum reference and value attributes of DataSetDiscoveryMetadata are Not equal to EPSG:4326 (WGS 84).	horizontalDatum reference and value are not EPSG 4326	Set the horizontalDatum reference and value attributes to EPSG 4326	PS 8.1, 8.2 and 14.2.1

118	If the file names in an exchange set are not in accordance with the Product Specification.	File names are not in accordance with the Product Specification.	Amend file names.	PS 11.9
119	For each feature instance that does not OVERLAP OR is WITHIN the bounding box.	Object outside area of coverage.	Remove object or amend coverage.	
120	For each feature instance which does not have a valid feature class label/code as defined by the feature catalogue.	Object has invalid feature class code.	Amend object class code.	Logical consistency
121	For each attribute which does not have a valid attribute label/code as defined by the feature catalogue.	Attribute has invalid attribute label/code.	Amend attribute label/code.	Logical consistency
122	For each feature object which contains attributes outside the list of permissible attributes for the feature class (as defined in the feature catalogue).	Attribute not permitted on feature class.	Remove attribute.	Logical consistency
123	If the order of the data in a dataset is not correct.	Incorrect data order.	Amend data order.	PS 10.7.1
124	For each attribute instance where the total number of instances exceed the permitted number of instances	Too many instances of attribute.	Ensure correct attribute encoding.	Logical consistency
125	For each feature instance where fixedDateRange subattributes dateEnd and dateStart are notNull AND their values are identical.	Object has identical values of periodicDateRange subattributes dateEnd and dateStart.	Ensure values of periodicDateRange subattributes dateEnd and dateStart are logical.	Logical consistency
126	For each feature instance where fixedDateRange subattribute dateStart is notNull AND dateEnd is Null OR not Present.	Object has dateStart without a value of dateEnd.	Populate dateEnd or remove dateStart.	Logical consistency
127	For each feature instance where fixedDateRange subattribute is notNull AND dateStart is Null OR not Present.	Object has dateEnd without a value of dateStart.	Populate dateStart or remove dateEnd.	Logical consistency
128	For each linear geometry which contains vertices at a density Greater than 0.3mm at 1:10000.	Vertex density exceeds the allowable tolerance.	Generalise edge(s).	PS 6.1

Commented [ME2]: Need to find a reference for this.

Annex A

1.0 Introduction

1.1 ISO 19125-1:2004 geometry.

This clause defines ISO 19125-2004 geometric terms used in this Annex.

1.1.1 Definitions for ISO 19125-1:2004 geometry

Note that these definitions are for the primitives defined by ISO 19125-1:2004 which are single point, single line, and single area geometry objects.

- *Polygon* – A Polygon has a geometric dimension of 2. It consists of a boundary and its interior, not just a boundary on its own. It is a simple planar surface defined by 1 exterior boundary and 0 or more interior boundaries. The geometry used by an S-57 Area feature is equivalent to a Polygon.

- *Polygon boundary* – A Polygon boundary has a geometric dimension of 1 and is equivalent to the outer and inner rings used by an S-57 Area feature.

- *LineString* – A LineString is a Curve with linear interpolation between Points. A LineString has a geometric dimension of 1. It is composed of one or more segments – each segment is defined by a pair of points. The geometry used by an S-57 Line feature is equivalent to a LineString.

- *Line* - An ISO 19125-1:2004 line is a LineString with exactly 2 points. Note that the geometry used by an S-57 Line feature is equivalent to a LineString, not a line in ISO 19125-1:2004 terms. In this document the term Line refers to an S-57 Line feature or a LineString which can have more than two points.

- *Point* – Points have a geometric dimension of 0. The geometry used by an S-57 Point feature is equivalent to an ISO 19125-1:2004 point.

- *Reciprocal* – inversely related or opposite.

The following table matches 19125-1:2004 geometric terms to S-57 terms:

ISO 19125-1:2004	S-57
Polygon	Area feature geometry OR Area
Polygon boundary	Outer and inner rings
LineString	Line feature geometry OR Line
Point	Point feature geometry OR Point

1.1.2 Definition of symbols used in ISO 19125-1:2004

I = interior of a geometric object
E = exterior of a geometric object
B = boundary of a geometric object
∩ = the set theoretic intersection
U = the set theoretic union
∧ = AND
Ú = OR
≠ = not equal
∅ = the empty or null set
a = first geometry, interior and boundary (the topological definition)
b = second geometry, interior and boundary (the topological definition)
dim = geometric dimension – 2 for Polygons , 1 for LineStrings, and 0 for Points

Dim(x) returns the maximum dimension (-1, 0, 1, or 2) of the geometric objects in x, with a numeric value of -1 corresponding to dim (Æ).

Note:
- Neither interior nor exterior include the boundary (i.e. I, E and B are mutually exclusive).
- The boundary of a Polygon includes its set of outer and inner rings.
- The boundary of a LineString is its end points except for a closed LineString, which has no boundary; the rest of the LineString is its interior.
- A Point does not have a boundary.

1.2 ISO 19125-1:2004 geometric operator relationships

In ISO 19125-1:2004 (see Reference [1]), the dimensionally extended nine-intersection model (DE-9IM) defines 5 mutually exclusive geometric relationships between two objects (Polygons, LineStrings, and/or Points). One and only one relationship will be true for any two given objects (see Reference [2]):

1. WITHIN
2. CROSSES
3. TOUCHES
4. DISJOINT
5. OVERLAPS

There are others that help further define the relationship:

1. CONTAINS
- the reciprocal of WITHIN
- Within is the primary operator; however, if **a** is not within **b** then **a** may contain **b** so CONTAINS may be the unique relationship between the objects.

2. EQUAL
- a special case of WITHIN / CONTAINS.

3. INTERSECTS
- reciprocal of DISJOINT
- have at least one point in common

4. COVERS and is COVERED_BY
- reciprocal operators
- extends CONTAINS and WITHIN respectively

5. COINCIDENT

Note that COVERS, COVERED_BY, and COINCIDENT relational operators are not described in the ISO 19125-1:2004 document.

The formulas given in this annex (e.g. a.Disjoint(b) $\Leftrightarrow$ a $\cap$ b = $\emptyset$) are the generalized ones given for ISO 19125-1, not the more specific DE-9IM formulas (i.e. DE-9IM predicates). The generalized formulas use topologically closed notation (i.e. geometry includes the interior and boundary unless otherwise stated), whereas the DE-91M formulas refer to the interior and boundary of geometry separately. Note that different versions of documents describing 19125-1 give different generalized formulas – this annex is using the formulas that are the most consistent with the DE-9IM predicates. If a generalized formula appears to contradict a DE-9IM predicate as defined in ISO 19125-1:2004, the DE-9IM predicate takes precedence. Software is expected to be consistent with DE-9IM predicates.

1.3 How the relationships apply to S-57 Features

Geometric relationships will be tested on an entire S-57 feature object as a single geometric entity. Note that S-57 Point, Line, and Area feature geometry is equivalent in ISO 19125-1:2004 terms to Point, LineString, and Polygon geometry respectively.

A Line feature in S-57 may be made up of several individual edges. The geometric relationship operators used with a Line feature will consider the sequence of edges as a single geometry (LineString).

A test on an Area feature will operate on the entire Polygon.

In an S-57 file a Line or Area feature may be split into pieces as a result of a cutting operation from a data source. In that case each feature record in the dataset is treated as a separate LineString or Polygon when testing geometric relationships.

If a test intends to operate only on a feature's specific components – Polygon boundary (all rings), Polygon outer ring, Polygon inner rings, edges, vertices, or nodes then it must make this explicit in the description of the test. When a specific linear portion is specified in a test (Polygon boundary, edge) then it is treated as a LineString while individual vertices or points will be treated as points.

For example a test to look for cases where object class A OVERLAPS object class B would operate on the entire geometry. While a test to see if boundary of Area object class A OVERLAPS an edge of Line class B will be comparing Area boundaries to edges using Line to Line comparisons.

2.0 Geometric Operator Definitions

The ISO 19125-1 definitions referenced in this clause refer to clause 6.1.14.3 entitled "Named spatial relationship predicates based on the DE-9IM" in the ISO 19125-1:2004 document.
(In the diagrams within this annex LineString corresponds to the S-57 Line geometric primitive)

EQUALS – Geometric object **a** is spatially equal to geometric object **b**.
The two geometric objects are the same. This is a special case of WITHIN.

Examples of the EQUALS relationship

Note: ISO 19107:2003 describes equality more formally as:

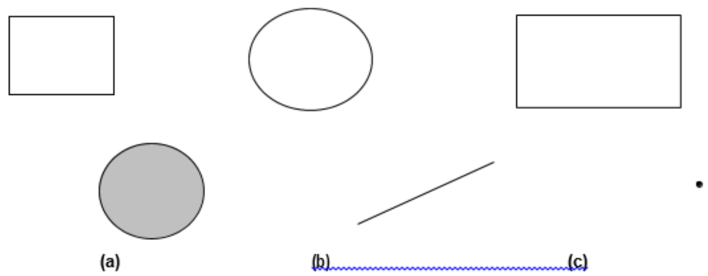
> Two different GM_Objects are equal if they return the same Boolean value for the operation GM_Object::contains for every tested DirectPosition within the valid range of the coordinate reference system associated to the object.

NOTE Since an infinite set of direct positions cannot be tested, the internal implementation of equal must test for equivalence between two, possibly quite different, representations. This test may be limited to the resolution of the coordinate system or the accuracy of the data. Application schemas may define a tolerance that returns true if the two GM_Objects have the same dimension and each direct position in this GM_Object is within a tolerance distance of a direct position in the passed GM_Object and vice versa.

For the purposes of S-127 Validation Checks, a GM_Object is any spatial object as described in A.1.1 (Polygons, LineStrings, and Points). A spatial object is always equal to itself, i.e., **a** EQUALS **a** is always true.
DISJOINT – Geometric object **a** and geometric object **b** do not intersect.
The two geometric objects have no common points.
The ISO 19125-1 definition of DISJOINT is:

$$\mathbf{a}.Disjoint(\mathbf{b}) \Leftrightarrow \mathbf{a} \cap \mathbf{b} = \emptyset$$

This translates to: **a** is disjoint from **b** if the intersection of **a** and **b** is the empty set.



Examples of the DISJOINT relationship

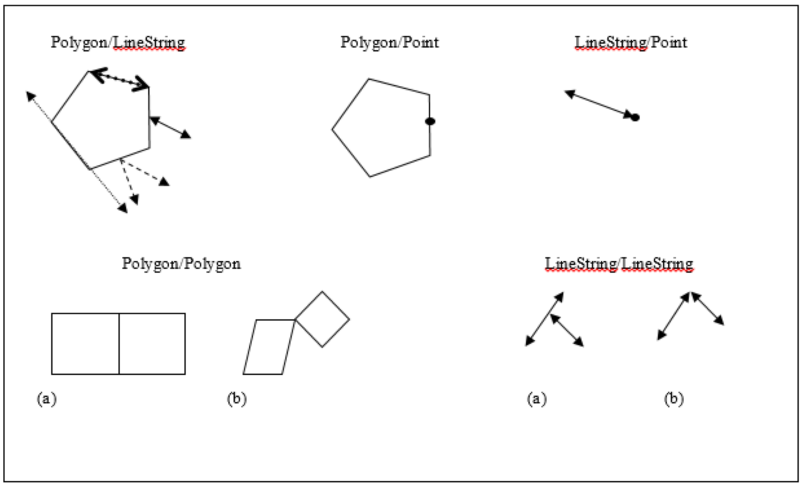
TOUCHES – Geometric object **a** intersects with geometric object **b** but they do not share interior points.
Only the boundary of one geometry intersects with the boundary or interior of another geometry.
The only thing the geometric objects have in common is contained in the union of their boundaries.

The ISO 19125-1 definition of TOUCHES is:

$$\mathbf{a}.Touch(\mathbf{b}) \Leftrightarrow (I(\mathbf{a}) \cap I(\mathbf{b}) = \emptyset) \wedge (\mathbf{a} \cap \mathbf{b}) \neq \emptyset$$

This translates to: **a** touches **b** if the intersection of the interior of **a** and the interior of **b** is the empty set AND the intersection of **a** and **b** is not the empty set.

Note: This operator applies to the Area/Area, Line/Line, Line/Area, Point/Area, and Point/Line relationships. It does not apply to a Point/Point relationship since points do not have a boundary.



Examples of the TOUCHES relationship.

Note the Polygon touches Polygon example (a) is also a case where the Polygon boundaries are COINCIDENT. In the Polygon/LineString example two of the LineStrings that share a linear portion of the Polygon boundary are also COINCIDENT with the Polygon boundary.

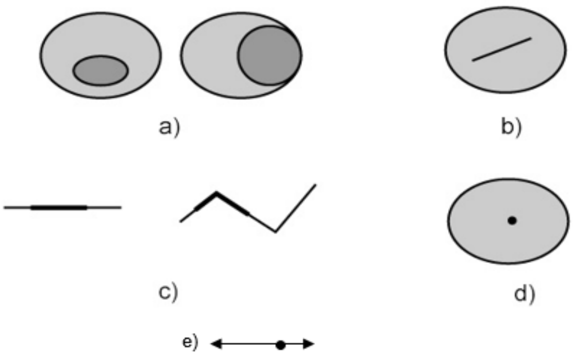
WITHIN – Geometric object **a** is completely contained in geometric object **b**.
WITHIN includes EQUALS.

The definition of WITHIN is:

$\mathbf{a}.\,\text{Within}(\mathbf{b}) \Leftrightarrow (\mathbf{a} \cap \mathbf{b} = \mathbf{a}) \wedge (I(\mathbf{a}) \cap I(\mathbf{b}) \neq \emptyset)$

This translates to: **a** is within **b** if the intersection of **a** and **b** equals **a** AND the intersection of the interior of **a** and the interior of **b** is not the empty set.

Note that this formula matches the one given in **the OpenGIS Simple Features Specification for SQL, Revision 1.1 (OpenGIS Project Document 99-049, Release Date: May 5, 1999**) which is the precursor to ISO 19125-1.



Examples of the WITHIN relationship — Polygon/Polygon (a), Polygon/LineString (b), LineString/LineString (c), Polygon/Point (d), and LineString/Point (e)

Note that a Line that completely falls on a Polygon boundary is not WITHIN the Polygon, it TOUCHES it. In that case it would also be COINCIDENT with the Polygon boundary and COVERED_BY the Polygon.

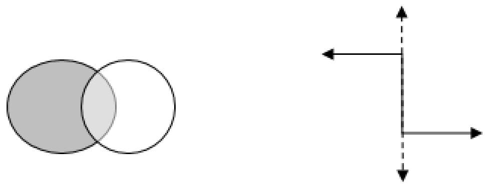
OVERLAPS - The intersection of two geometric objects with the same dimension results in an object of the same dimension but is different from both of them.
For two Polygons or two LineStrings, part of each geometry, but not all, is shared with the other.

The OVERLAPS relationship is defined for Area/Area and Line/Line relationships. Points are either equal or disjoint.
Note that this does not include lines that cross.

The ISO 19125-1 definition of OVERLAPS is:

$$\boldsymbol{a}.Overlaps(\boldsymbol{b}) \Leftrightarrow (dim(I(\boldsymbol{a})) = dim(I(\boldsymbol{b})) = dim(I(\boldsymbol{a}) \cap I(\boldsymbol{b}))) \wedge (\boldsymbol{a} \cap \boldsymbol{b} \neq \boldsymbol{a}) \wedge (\boldsymbol{a} \cap \boldsymbol{b} \neq \boldsymbol{b})$$

This translates to: **a** OVERLAPS **b** if the geometric dimension of:
(1) the interior of **a**
(2) the interior of **b**
(3) the intersection of the interiors of **a** and **b**
are all equal AND the intersection of **a** and **b** does not equal either **a** or **b**.



Examples of the OVERLAPS relationship

Note Lines that OVERLAP are also COINCIDENT.

CROSSES – The intersection of geometric object **a** and geometric object **b** returns geometry with a dimension less than the largest dimension between **a** and **b** but is not the same as geometric object **a** or **b**.

Two LineStrings cross each other if they meet on an interior point. A LineString crosses a Polygon if the LineString is partly inside the Polygon and partly outside.

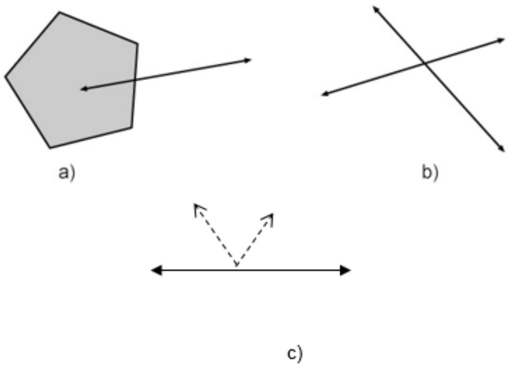
The definition of CROSSES is:

$$\boldsymbol{a}.Cross(\boldsymbol{b}) \Leftrightarrow (I(\boldsymbol{a}) \cap I(\boldsymbol{b}) \neq \emptyset) \wedge (dim(I(\boldsymbol{a}) \cap I(\boldsymbol{b})) < max(dim(I(\boldsymbol{a})), dim(I(\boldsymbol{b})))) \wedge (\boldsymbol{a} \cap \boldsymbol{b} \neq \boldsymbol{a}) \wedge (\boldsymbol{a} \cap \boldsymbol{b} \neq \boldsymbol{b})$$

This translates to: **a** crosses **b** if the intersection of the interiors of **a** and **b** is not the empty set AND the dimension of the result of the intersection of the interiors of **a** and **b** is less than the largest dimension between the interiors of **a** and **b** AND the intersection of **a** and **b** does not equal either **a** or **b**.

Note that "$(I(\boldsymbol{a}) \cap I(\boldsymbol{b}) \neq \emptyset) \wedge$ " was added to the beginning of the ISO 19125-1 formula so that it would not be true for disjoint geometry.

The CROSSES operator only applies to Line/Line and Line/Area relationships.



Examples of the CROSSES relationship

Note that example c) shows one solid line and one dashed line – their interiors intersect. If any Line were split into two separate Line features at the intersection point then the relationship would be TOUCHES because a boundary would be involved.

INTERSECTS is the reciprocal of DISJOINT.
The two geometric objects cross, overlap or touch, or one is within (or is contained by) the other. They have at least one common point.

CONTAINS is the reciprocal of WITHIN.
*Given two geometric objects, **a** and **b**, if **a** is within **b** then **b** must contain **a**.*

COVERED_BY (not a standard ISO 19125-1 operator)

No point of geometry **a** is outside geometry **b**.

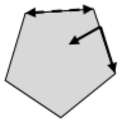
The definition of COVERED_BY is:
a. COVERED_BY (**b**) ⇔ (**a** ∩ **b** = **a**)

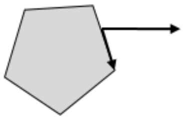
This translates to: **a** is COVERED_BY **b** if the intersection of **a** and **b** equals **a**.

The following expressions are equivalent to **a** is COVERED_BY **b**:

1. Polygon (**a**) is COVERED_BY Polygon (**b**): Polygon **a** is WITHIN a polygon **b** (WITHIN includes EQUALS)
2. Point (**a**) is COVERED_BY Polygon (**b**): Point **a** is WITHIN or TOUCHES polygon **b**
3. Line (**a**) is COVERED_BY Polygon (**b**): Line **a** is WITHIN polygon **b** or WITHIN the boundary of Polygon **b**
4. Line (**a**) is COVERED_BY Line (**b**): Line **a** is WITHIN Line **b** (WITHIN includes EQUALS)
5. Point (**a**) is COVERED_BY Line (**b**): Point **a** is WITHIN or TOUCHES Line **b**
6. Point (**a**) is COVERED_BY Poiint (**b**): Point **a** EQUALS Point **b**

Note that the figure below on the left is an example of Lines that are COVERED_BY a polygon.
The figure on the right is NOT an example of a Line that is covered by a Polygon – it is an example of a Line that TOUCHES a Polygon. In both cases the Lines are COINCIDENT with the Polygon boundary.





Examples of COVERED_BY and NOT COVERED_BY

COVERS (not a standard ISO 19125-1 operator)

COVERS is the reciprocal of COVERED_BY.

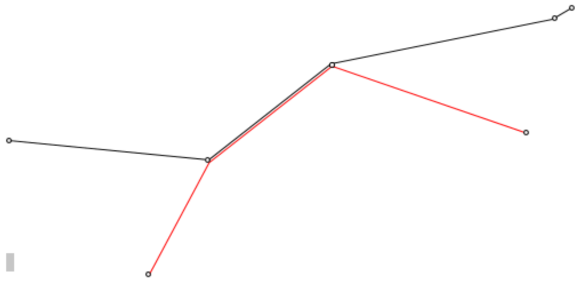
*Given two geometric objects, **a** and **b**, if **a** is* COVERED_BY ***b** then **b** must cover* **COINCIDENT** (not an ISO 19125-1 operator)

Two geometric Lines OVERLAP or one geometric Line is WITHIN the other. Note that EQUAL Lines are also COINCIDENT by this definition.
The intersection of two geometric Lines results in one or more Lines.

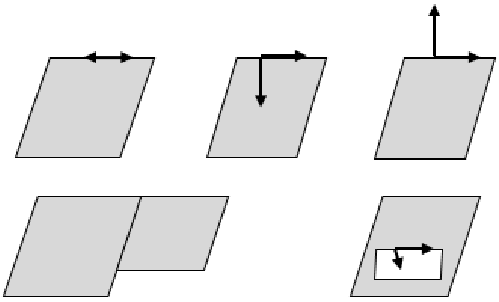
This operator is only to be used to compare a Line with another Line. Note that normally the boundary of a Polygon is not the same as a Line but for this operation the boundary of a Polygon, exterior and interior rings, is treated as Lines for the COINCIDENT test.

The following expressions are equivalent to **a** is COINCIDENT with **b**:

1. Polygon (**a**) is COINCIDENT with Polygon (**b**): The boundary of Polygon **a** OVERLAPS or is WITHIN the boundary of Polygon **b**.
2. Line (**a**) is COINCIDENT WITH Polygon (**b**) : Line **a** OVERLAPS or is WITHIN the boundary of Polygon **b**.
3. Line (**a**) is COINCIDENT WITH Line (**b**): Line **a** OVERLAPS or is WITHIN Line **b**



Example of two COINCIDENT geometric LINES



Examples of COINCIDENT objects

Above are other examples of objects COINCIDENT with the boundary of a Polygon. LineStrings following a portion of a Polygon boundary or Polygons sharing a boundaryportion.
Note that by definition a Line can be COINCIDENT with an interior boundary of a Polygon.
Note that other relationships may also be true, such as COVERED_BY or TOUCHES, since COINCIDENT is not mutually exclusive.

Bibliography

[1] ISO 19125-1:2004, *Geographic Information – Simple feature access – Part 1 Common architecture*

[2] CLEMENTINI, E., DI FELICE, P., VAN OOSTROM, P. *A Small Set of Formal Topological Relationships Suitable for End-User Interaction, in D. Abel and B. C. Ooi (Ed.), Advances in Spatial Databases* — Third International Symposium. SSD 1993. LNCS **692**, pp. 277-295. Springer Verlag. Singapore (1993)

[3] ISO 19107:2003, *Geographic information* | *Spatial schema*

[4] OpenGIS Simple Features Specification for SQL, Revision 1.1 *(OpenGIS Project Document 99-049, Release Date: May 5, 1999)*