# Interchange of Spatial Data – Inhibiting Factors

Rodney James Thompson
Delft University of Technology
Delft, The Netherlands
rodnmaria@gil.com.au

Peter van Oosterom
Delft University of Technology
Delft, The Netherlands
oosterom@otb.tudelft.nl

**MAJOR THEME: Interoperability.**
**NATURE OF THE ABSTRACT: Scientific**

## SUMMARY

For many years now, the interchange of textural and numeric data between hardware platforms and database management systems has been fairly simple and error free. As a result, managers and information technology professionals have been unpleasantly surprised to find that this is not the case with spatial data.

There is an expectation that, having created a database and populated it with carefully validated data (including spatial features), it should be possible to supply it to another party. The shock comes when the second party (customer) finds that some features that have been supplied are not valid.

This paper explores some of the issues that lead to this situation, and suggests an interim pragmatic solution. Intrinsic to this discussion, the issue of data transfer accuracy is addressed.

**KEYWORDS:** *Spatial Data Interchange, Validation, Standardization, Accuracy.*

## INTRODUCTION

In order to make the problem more identifiable, two specific cases are introduced:

### Case 1

An officer of an organization which supplies Cadastral data was addressing a meeting of users of that data. A question was addressed from the floor "Why do you not validate your data before you release it". A representative example was investigated and found to have been correct in the suppliers database (***Figure 1***), but due to inaccuracy in the transfer, a small movement of points had caused what the user refers to as a "Butterfly polygon" (a self intersecting polygon – ***Figure 2***).

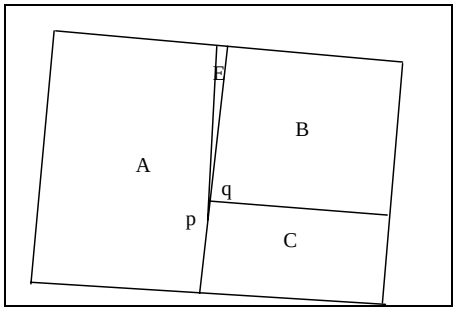


***Figure 1:*** Cadastral data in the source database

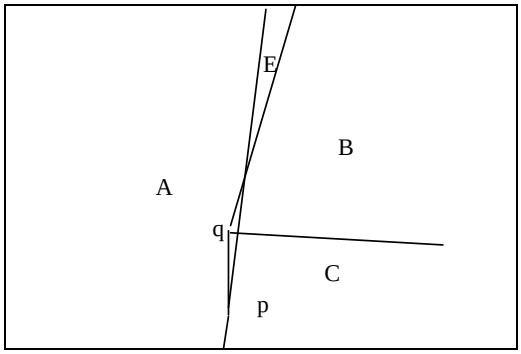


***Figure 2:*** The same data on arrival (detail)

### Case 2

Data was being loaded into a large, primarily topographic, database. The message "Geometry is Self-Intersecting" was displayed against a particular feature. This astonished the person who supplied the data since it had been validated immediately before the attempt to load it, using the validation routines of a major software vendor. Investigation showed that the validation had been carried out in MGA94 (Mapping Grid of Australia 1994) (Geoscience Australia 2005), whereas the target database required GDA94 (Geocentric Datum of Australia). The error arose because the conversion caused a very small relative movement of points within the feature.

## CURRENT DATA TRANSFER PROTOCOLS

There are a plethora of transfer formats currently in use, but they fall into two main categories: the ASCII (readable character) and the binary formats. Since almost all transfer formats fall into one or other of these categories, a few examples of each can be used to illustrate the issues that apply to the remainder. This paper will use as examples, Shapefile (ESRI 1997) and Binary XML (OGC 1993) as examples of binary formats, and GML 2 (OGC 2002) as an example of ASCII format.

### Object Definition Issues

In the past, but to a lesser degree today, the actual geometric object definitions could differ markedly from one storage environment to another (van Oosterom 2003). This has been partially solved by the wide adoption of the OGC Simple Feature Specification (OGC 1999) and associated specifications, but some issues still stand. These issues apply equally to the ISO 19107 Spatial Schema (ISO-TC211 2001).

### Definition of Equality

The ISO 19107 Spatial Schema definition and the OGC Simple Feature specification both allow the use of a tolerance in the determination of equality. This tolerance is left to the implementer, both in magnitude and in method of application. As a result, features which are equal within one implementation may be unequal within another. Further, "equality" defined on this basis is not an equivalence relation (Thompson 2005).

### Definition of Validity

ISO 19107 defines the isSimple test, and this is treated as a test for validity by the OGC specification. By contrast with the equality test, no tolerance is allowed for this test. As a result, it is possible for features to be marginally valid – such that a very small perturbation in the relative positions of points can cause a valid feature to become invalid (Thompson 2005).

### Binary Format Issues

These protocols allow the transfer of geographic positions as ordered pairs of either floating point or integer numbers. In either case, provided the representation is identical in all respects in the source and the target environments, there will be no change to any point positions.

Unfortunately, the actual storage mechanisms for geometric objects within spatial database structures are proprietary, usually not under the control of the data custodians, and often not published. There is a strong chance that where the source and target environments are not from the same supplier, they will not be identical. As was shown by (van Oosterom 2003), the vendors' interpretation of the OGC specification is not always consistent.

Even in the case of identical environments on the source and target environments, and using the OGC specification, the same "Spatial Reference" may not be in use. For example, the supplier could be storing in Latitude/Longitude form, while the target could be using a transverse mercator projection grid system.

Thus, in all but a minority of cases, there will be some small relative positional movement of points in the data.

### ASCII Format Issues

The above issues also apply to the ASCII formats, but in addition, these formats encode the geographical positions as ordered pairs of decimal numbers of a finite precision. If the precision of the transfer encoding is less than the precision of the source data, then at the destination the geographic position will not be identical to that at source.

## RESULTS OF RANDOM RELATIVE MOVEMENTS

Any of the above factors may result in small random relative movements of the points in transit, which has the potential to introduce topological failures. For example In ***Figure 3***, it may be a topological validity requirement that point *p* be below the line AB. If the distance from *p* to AB is small, then rounding may introduce slight movements to the line and the point, causing the constraint to be violated.

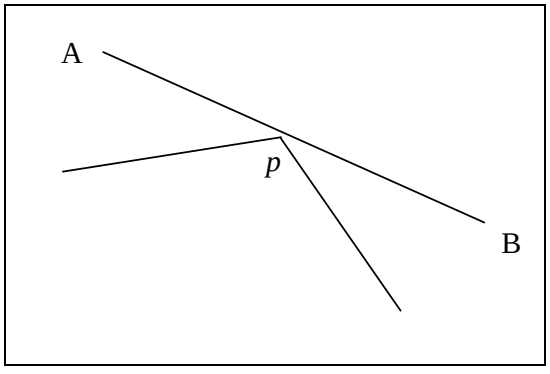


***Figure 3:*** Example of a point near an unrelated line

In the new situation (shown in ***Figure 4*** in full lines), the line AB is cut by the lines through *p*, leading to a topological failure. In an ideal world, this would not happen of course, since no point would be placed so close to a line as to be indistinguishable from it within the accuracy of the data.

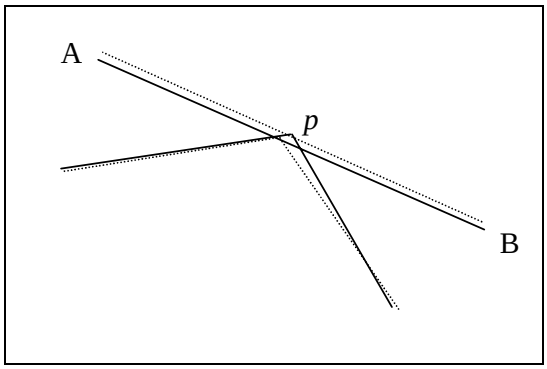


***Figure 4:*** Above example after small perturbation

Crossovers of this kind do occur however, and with surprising frequency. Unfortunately the original digitising and other processes can generate small "knots" in the linework, some of which do not, in the first instance, fail validation because they do not result in lines crossing. They are not visible at any sensible scale, and so will be highly likely to be overlooked by the operators and allowed into the database. In ***Figure 5***, both objects are actually incorrect, since they contain points which are far too close together in relation to the accuracy of the data. However only the object on the left is detected as invalid. Any operation on the data which allows random relative movement may cause topology breakdowns.

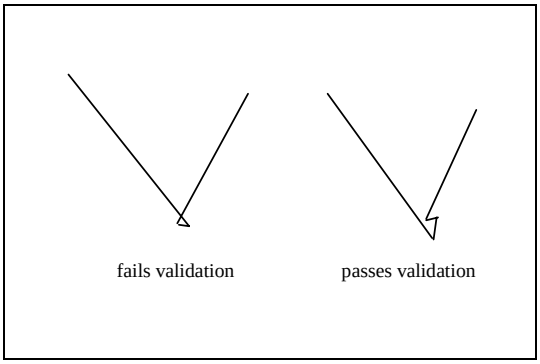


***Figure 5:*** "Knots" in linework

It might be thought that increasing the resolution used for storing the data would alleviate the problem, but paradoxically, the finer the resolution of the data, the more difficult is the problem to locate and correct. Finer resolution means that the knots may be far smaller while still causing trouble, and smaller movement during the transmission are likely to create a failure. Further, transferring more digits than necessary can be a costly exercise.

## HOW ACCURATE A TRANSFER MECHANISM IS NEEDED?

A data set which is of similar accuracy to the information in a 1:250000 scale paper map needs a minimum 3 digits after the decimal point if interchanged in degrees of latitude and longitude.

In fact, any map-sized region would be adequately represented in 4 significant figures if a local origin is used. For example, if a 1:25000 map is 1/16 of a degree square and its origin is recorded as (for example) 26.17S, 153.72E (being a point within or near its area), the individual points on its surface could be recorded as pairs of 4 digit numbers. Thus coordinates (1023, 6792) could represent point (26.171023S, 153.6792E).

When transmitting data at 1:250000 scale, there is really no information in the digits past the first 3 after the decimal point. At 1:25000 - 4 digits, and at 1:2500 5 digits carry the information. Currently GML files are often transferred with 9 or more digits following the decimal point. This equates to a clearly implausible "accuracy" of about 0.1 mm at ground scale. The main reason for the additional digits is to reduce the relative point movements – as described above.

## WHY DOES THIS MATTER?

The GML with 9 digits after the decimal point does not compress well for transmission. For example, a GML file of 2.4 mB compresses to 0.6 mB (a 4:1 reduction) (***Figure 6*** and ***Table 1*** – GML 9). If restricted to just 5 digits after the decimal point it reduces to 1.8mB, compressing to 0.3mB (a 5.8:1 reduction) (***Figure 6*** and ***Table 1*** – GML 5). Not only has the original file been reduced in size, but the compression ratio has been improved.

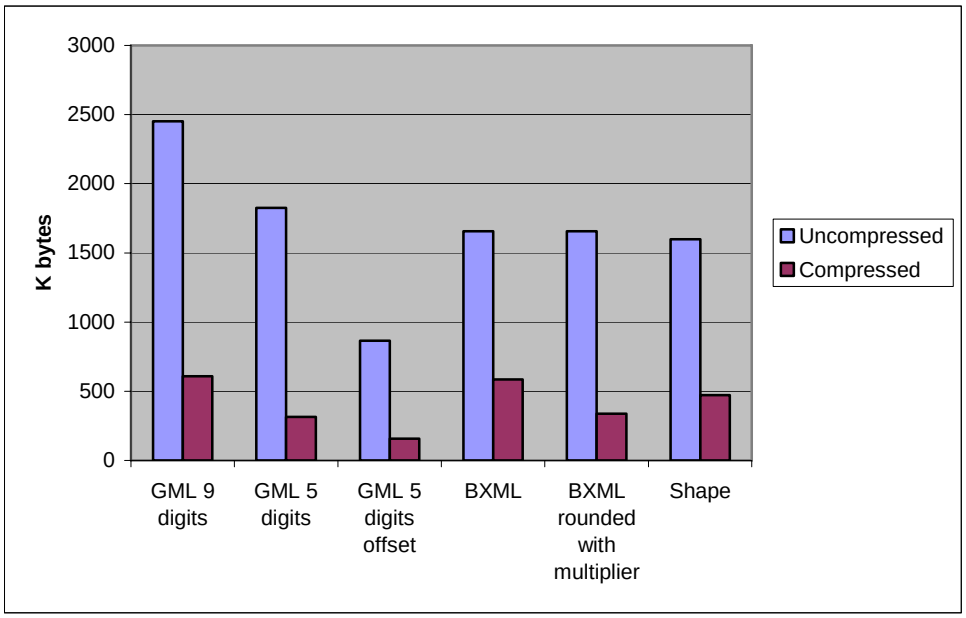


***Figure 6:*** Compression of GML and variants (see ***Table 1***)

| | Initial Size | Zipped Size | Percentage | Percent of GML size |
|---|---|---|---|---|
| GML 9 digits | 2451 | 608 | 24.8 | 24.8 |
| GML 5 digits | 1824 | 314 | 17.2 | 12.8 |
| GML 5 digits offset | 864 | 158 | 18.3 | 6.4 |
| BXML | 1656 | 585 | 35.3 | 23.9 |
| BXML rounded with multiplier | 1656 | 337 | 20.4 | 13.7 |
| Shape | 1599 | 473 | 29.6 | 19.3 |

***Table 1:*** Compression ratios of GML and variants

There is a good reason for this. Spurious digits, such as those generated by using a higher accuracy than necessary are "pseudo random" - it is not possible to distinguish them from a random sequence. This means that they will not compress well. A string of random digits will require nearly 4 bits per digit in any compression scheme, no matter how efficient.

The Open GIS Consortium has developed a "Binary XML" (OGC 2003) specification to address the problem, but in practice, it proved to be disappointing in transmitting real data (BXML in ***Figure 6***). This is not really surprising, since the lat/lon values are encoded as pairs of 64bit floating point numbers with the equivalent of 15 significant decimal digits, while as described above, only the first few are meaningful. This means that the majority of bits of the encoded value are "pseudo random", and a pseudo random string of bits cannot be compressed at all. Acceptable results can only be achieved for binary XML if the coordinate values are multiplied by a scaling factor and truncated to integers ("BXML rounded with multiplier" in ***Figure 6***).

The big improvements that OGC claim probably come from using "new" data. If a collection of features is encoded or digitised to a particular accuracy, the latitude/longitude values will consist of mainly zero bits, which do compress well. For example, the number 17.5, will be stored with an 11 bit exponent, followed by the bit pattern 100011000…, where the last 1 is followed by 47 zeroes (Goldberg 1991). In general, where there are $n$ significant digits in the number, with appropriate normalization, the final $b$ bits will be zero, where $b=53-n*3.3$ (approximately). Note also that the Shape format compression (in ***Figure 6***) is comparable with the Binary GML format, as would be expected.

But, suppose that data is now passed through a transformation such as change of datum. The arithmetic operations will generate new binary values for the latitude and longitude, and these values will have pseudo random bits in place of the zeroes. It is not usual to round results when applying such changes, so the compression ratios would change dramatically.

## POSSIBLE SOLUTIONS

The solution suggested by the rhetorical question posed by the audience member in Case 1 might be considered – to convert the data into transfer format, and then validate it using the proposed client's software. This is impractical, since it requires a data supplier to purchase software to match that used by any customer to whom they wish to supply.

The next solution could be the *caveat emptor* approach – that "The data was valid when it left our database". This is the most commonly used approach, but can lead to embarrassment as in Case 1.

The ideal solution to this issue would be for a rigorous definition of geometric objects to be specified, which is independent of any Vendor software, and where the behaviour is completely and prescriptively defined. Several approaches are currently in train, such as the "Realms" Approach (G*ü*ting 1993), "Dual Grid" (Lema 2002), "Constraint Programming" (Kanellakis 1995) and the

"Regular Polytope" (Thompson 2005). None of these are currently available in any commercial software, so an interim solution is needed.

## AN INTERIM PRAGMATIC SOLUTION

Milenkovic defines a set of normalization rules, based on a parameter ε which is chosen so that the "distance between a point and a line segment can be calculated with accuracy $\frac{1}{10}\varepsilon$" (Milenkovic 1988 page 382). Two of the rules are:

- No two vertices are closer than ε.
- No vertex is closer than ε to an edge of which it is not an endpoint.

For spatial data that satisfies these rules, a random relative movement of points which does not exceed ε will not result in an invalid geometry (by the OGC definition of "valid"). On the other hand, on arrival the data may no longer be valid by the Milenkovic rules, since the vertices and edges may have moved closer together than ε.

## THE ROBUSTNESS PARAMETER

The solution proposed here recognises that there is a distinction between well defined features, and features which are "fragile" – that are likely to become invalid if subjected to small perturbations. It is suggested that, rather than defining a Boolean operation ("isValid()" or "isSimple()" etc.), a method of calculating a "robustness" parameter ρ be defined such that the movement of all points by a distance of < ρ in any direction will guarantee to leave the object valid. A large robustness value would indicate a "robust" representation, while a small value should be a warning of potential problems.

In the context of "Design by Contract" (Mitchell 2002), if an operation (a data transfer, datum change, etc.), could introduce an inaccuracy, it should contract for a region with features of a sufficiently large robustness value. For example, if an operation is only accurate to 1mm, given a set of features with ρ=0.3mm, a valid output cannot be contracted.

This robustness parameter can be calculated in the same way as the Milenkovic normalization is checked.

> Let $\rho_1$ be the minimum distance between any two points.
>
> Let ρ2 be the minimum distance between any point and any line.
>
> Let ρ = min(ρ1 , ρ2 ).

That is to say, ρ is the largest number for which the geometry is Milenkovic normal with ε=ρ.

For each feature, the robustness parameter is recorded. When data is to be transferred to a user, the user's tolerance requirement, ρ' is determined, and the accuracy of the available transport mechanism, δ is determined. If ρ'+ δ < ρ, a transfer may be contracted. Provided that the target environment has a compatible definition for the spatial primitive, and the same definition of validity, the transmission can be guaranteed correct.

## CONCLUSION

A fully rigorous approach to the transfer of spatial data is some years in the future. Until that time, it will be necessary to note carefully the accuracy of the data, the robustness of the feature encoding, and the resolution of the transfer mechanism, and to ensure that these three items are compatible. It is also important to be aware of possible differences between the geometric concepts at the source and target repositories. Referring back to cases 1 and 2, both would have been recognised as being fragile and therefore unsuitable for supply, given the available transfer accuracy.

ESRI, 1998 *ESRI Shapefile Technical Description* Environmental Systems Research Institute Inc., http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

Geoscience_Australia, 2005 *Geocentric Datum of Australia [GDA]*, Australian Government, Canberra http://www.ga.gov.au/geodesy/datums/gda.jsp.

Goldberg, D., What Every Computer Scientist Should Know About Floating-Point Arithmetic, *Computing Surveys*. 1991

Güting, R. H. and Schneider, M., Realms: A foundation for spatial data types in database systems, *3rd International Symposium on Large Spatial Databases (SSD)*, Singapore. 1993

ISO-TC211, 2001 *Geographic Information - Spatial Schema*, International Organization for Standards, Geneva.

Kanellakis, P. C., Kuper, G. M. and Revesz, P. Z., Constraint query languages, in *Journal of Computer and System Sciences*, Vol 51 26-52. 1995

Lema, J. A. C. and Guting, R. H. Dual grid: A new approach for robust spatial algebra implementation, in *Geoinformatica*, Vol 6 **(**1), 57-76. 2002

Milenkovic, V. J., Verifiable implementations of geometric algorithms using finite precision arithmetic, in *Artificial Intelligence*, 377-401. 1988

Mitchell, R. and McKim, J. 2002, *Design by Contract, by Example*, Addison Wesley Professional.

OGC 1999, Open GIS Simple features Specification for SQL, Open GIS Consortium Inc, http://www.opengeospatial.org/specs/

OGC 2003, Geography Markup Language (GML) 2.1.2, Open GIS Consortium Inc. http://www.opengeospatial.org/specs/

OGC 2003, Binary-XML Encoding Specifications, Open GIS Consortium Inc. http://www.opengeospatial.org/specs/

Thompson, R. J. 2005, 3D Framework for Robust Digital Spatial Models, *Large-Scale 3D Data Integration.*, Zlatanova, S. and Prosperi, D., Eds., Taylor & Francis,.Boca Raton.

van Oosterom, P., Quak, W. and Tijssen, T. Polygons: The unstable foundation of spatial modeling, *International Society of Photogrammetry and Remote Sensing*, Quebec 2-3 October 2003.

## APPENDIX 1 SAMPLES OF DATA FORMATS

As examples of the data used in this study, *Tables 2 to 4* have been included as excerpts from the ASCII form of interchange formats. *Tables 5 and 6* are the implementations of binary XML used in this test, These have not been verified as correct, but it is believed that any anomalies in the encoding will not affect the conclusions drawn in this paper.

```
<gml:featureMember>
<rime:feature>
<rime:fType>shopping_centre</rime:fType>
<rime:fClass>feature</rime:fClass>
<rime:gType>PolyLine</rime:gType>
<rime:metaLink>meta1665.htm</rime:metaLink>
<rime:featId>3631217</rime:featId>
<rime:minCl>A</rime:minCl>
<rime:maxSCl>F</rime:maxSCl>
<rime:maxRCl>F</rime:maxRCl>
<rime:cLockNr>1665</rime:cLockNr>
<rime:as2482Code>1152</rime:as2482Code>
<gml:lineStringProperty>
<gml:LineString srsName="EPSG:4326">
<gml:coordinates>152.790301174,-27.616450116,0
152.790513337,-27.615246341,0 152.789773986,-27.615142941,0
152.789745584,-27.615304023,0 152.789573386,-27.615279943,0
152.789415047,-27.61617811,0  152.789591923,-27.616202861,0
152.789566457,-27.616347313,0 152.790301174,-27.616450116,0
</gml:coordinates></gml:LineString>
</gml:lineStringProperty>
</rime:feature>
</gml:featureMember>
```

***Table 2:*** GML 9 digit format (excerpt)

```
<LineString srsName="EPSG:4326">
<coordinates div="100000.0">152.7903,-27.61645,0
152.79051,-27.61525,0 152.78977,-27.61514,0
152.78975,-27.6153,0 152.78957,-27.61528,0
152.78942,-27.61618,0 152.78959,-27.6162,0
152.78957,-27.61635,0 152.7903,-27.61645,0
</coordinates>
</LineString>
```

***Table 3:*** GML 5 digit format (excerpt)

```
<LineString srsName="EPSG:4326">
<coordinates div="100000.0">
+15279030-2761645+0+21+120+0-74+11+0-2-16+0-18+2+0
-15-90+0+17-2+0-2-15+0+73-10+0
</coordinates>
</LineString>
```

***Table 4:*** GML 5 digit format with offset values

Note – in the 5 digit format examples above, the namespace qualifiers (“rime:” and “gml:”) have been omitted. This will slightly reduce the uncompressed file size, but it is not considered that the compressed file size will be affected sufficiently to invalidate the above argument. The data in *Table 4* is not an expression of the GML format, but is an example of a minimal ASCII representation.

```
00040360h: 00 34 41 30 00 34 41 30 00 34 02 06 02 1A 02 08 ; .4A0.4A0.4......
00040370h: 41 30 07 63 6F 6E 74 6F 75 72 34 02 09 41 30 41 ; A0.contour4..A0A
00040380h: 01 65 6C 65 76 61 74 69 6F 6E 5F 66 65 61 74 75 ; .elevation_featu
00040390h: 72 65 34 02 0A 41 30 09 50 6F 6C 79 4C 69 6E 65 ; re4..A0.PolyLine
000403a0h: 5A 34 02 15 41 30 02 34 30 34 02 0B 41 30 0C 6D ; Z4..A0.404..A0.m
000403b0h: 65 74 61 31 36 36 35 2E 68 74 6D 34 02 0C 41 30 ; eta1665.htm4..A0
000403c0h: 07 33 36 34 37 35 34 32 34 02 0D 41 30 01 41 34 ; .36475424..A0.A4
000403d0h: 02 0E 41 30 01 46 34 02 0F 41 30 01 46 34 02 10 ; ..A0.F4..A0.F4..
000403e0h: 41 30 04 31 36 36 35 34 02 11 41 30 04 35 30 30 ; A0.16654..A0.500
000403f0h: 32 34 02 16 03 17 05 04 FA 09 45 50 53 47 3A 34 ; 24......ú.EPSG:4
00040400h: 33 32 36 06 02 05 FB F9 40 63 19 1B 6F 30 1F BF ; 326...ûù@c..o0.¿
00040410h: C0 3B 93 F7 24 FE D6 CF 40 44 00 00 00 00 00 00 ; À;“÷$þÖÏ@D......
00040420h: 40 63 19 1D A5 DF C5 6F C0 3B 93 F6 82 82 14 B2 ; @c..¥ßÅoÀ;“ö,,.²
00040430h: 40 44 00 00 00 00 00 00 40 63 19 21 B5 60 2F D2 ; @D......@c.!µ`/Ò
00040440h: C0 3B 93 EA B8 81 4C 82 40 44 00 00 00 00 00 00 ; À;“ê¸LL,@D......
00040450h: 40 63 19 22 A1 FF DE 61 C0 3B 93 E1 86 BE DC B0 ; @c."¡ÿÞaÀ;“á†¾Ü°
00040460h: 40 44 00 00 00 00 00 00 40 63 19 22 D1 88 0C 42 ; @D......@c."Ñˆ.B
00040470h: C0 3B 93 DB 9C C0 B7 E3 40 44 00 00 00 00 00 00 ; À;“ÛœÀ·ã@D......
00040480h: 40 63 19 22 73 A8 18 62 C0 3B 93 D4 60 BF DE 52 ; @c."s¨.bÀ;“Ô`¿ÞR
00040490h: 40 44 00 00 00 00 00 00 40 63 19 1E 95 9F ED 79 ; @D......@c..•Ÿíy
000404a0h: C0 3B 93 B8 0E 01 7E 0D 40 44 00 00 00 00 00 00 ; À;“¸..~.@D......
000404b0h: 40 63 19 1D 1C 67 E2 09 C0 3B 93 AB 8D 80 0E 61 ; @c...gâ.À;“«□€.a
000404c0h: 40 44 00 00 00 00 00 00 40 63 19 1C 19 1F D9 06 ; @D......@c....Ù.
000404d0h: C0 3B 93 A6 46 7F 85 0A 40 44 00 00 00 00 00 00 ; À;“¦F□…@D......
000404e0h: 40 63 19 1A 87 DF CC 9E C0 3B 93 A1 00 80 AE 66 ; @c..‡ßÌžÀ;“¡.€®f
000404f0h: 40 44 00 00 00 00 00 00 40 63 19 19 0E 00 40 39 ; @D......@c....@9
00040500h: C0 3B 93 A2 F3 FE 72 95 40 44 00 00 00 00 00 00 ; À;“¢óþr•@D......
```

***Table 5:*** BXML format excerpt (note the pseudo random digits)

```
00040360h: 00 34 41 30 00 34 41 30 00 34 02 06 02 1A 02 08 ; .4A0.4A0.4......
00040370h: 41 30 07 63 6F 6E 74 6F 75 72 34 02 09 41 30 41 ; A0.contour4..A0A
00040380h: 01 65 6C 65 76 61 74 69 6F 6E 5F 66 65 61 74 75 ; .elevation_featu
00040390h: 72 65 34 02 0A 41 30 09 50 6F 6C 79 4C 69 6E 65 ; re4..A0.PolyLine
000403a0h: 5A 34 02 15 41 30 02 34 30 34 02 0B 41 30 0C 6D ; Z4..A0.404..A0.m
000403b0h: 65 74 61 31 36 36 35 2E 68 74 6D 34 02 0C 41 30 ; eta1665.htm4..A0
000403c0h: 07 33 36 34 37 35 34 32 34 02 0D 41 30 01 41 34 ; .36475424..A0.A4
000403d0h: 02 0E 41 30 01 46 34 02 0F 41 30 01 46 34 02 10 ; ..A0.F4..A0.F4..
000403e0h: 41 30 04 31 36 36 35 34 02 11 41 30 04 35 30 30 ; A0.16654..A0.500
000403f0h: 32 34 02 16 03 17 05 04 FA 09 45 50 53 47 3A 34 ; 24......ú.EPSG:4
00040400h: 33 32 36 06 02 05 FB F9 C0 D5 09 00 00 00 00 00 ; 326...ûùÀÕ......
00040410h: 40 E4 9B 20 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä› ....AN„€....
00040420h: C0 D5 02 40 00 00 00 00 40 E4 9B 40 00 00 00 00 ; ÀÕ.@....@ä›@....
00040430h: 41 4E 84 80 00 00 00 00 C0 D4 F6 00 00 00 00 00 ; AN„€....ÀÔö.....
00040440h: 40 E4 9D 80 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä□€....AN„€....
00040450h: C0 D4 F3 00 00 00 00 00 40 E4 9F 40 00 00 00 00 ; ÀÔó.....@äŸ@....
00040460h: 41 4E 84 80 00 00 00 00 C0 D4 F2 80 00 00 00 00 ; AN„€....ÀÔò€....
00040470h: 40 E4 A0 60 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä `....AN„€....
00040480h: C0 D4 F3 80 00 00 00 00 40 E4 A1 C0 00 00 00 00 ; ÀÔó€....@ä¡À....
00040490h: 41 4E 84 80 00 00 00 00 C0 D4 FF 80 00 00 00 00 ; AN„€....ÀÔÿ€....
000404a0h: 40 E4 A7 20 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä§ ....AN„€....
000404b0h: C0 D5 04 00 00 00 00 00 40 E4 A9 80 00 00 00 00 ; ÀÕ......@ä©€....
000404c0h: 41 4E 84 80 00 00 00 00 C0 D5 07 00 00 00 00 00 ; AN„€....ÀÕ......
000404d0h: 40 E4 AA 80 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä ª€....AN„€....
000404e0h: C0 D5 0B C0 00 00 00 00 40 E4 AB 80 00 00 00 00 ; ÀÕ.À....@ä«€....
000404f0h: 41 4E 84 80 00 00 00 00 C0 D5 10 40 00 00 00 00 ; AN„€....ÀÕ.@....
00040500h: 40 E4 AB 20 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä« ....AN„€....
00040510h: C0 D5 13 00 00 00 00 00 40 E4 A9 A0 00 00 00 00 ; ÀÕ......@ä© ....
00040520h: 41 4E 84 80 00 00 00 00 C0 D5 14 C0 00 00 00 00 ; AN„€....ÀÕ.À....
00040530h: 40 E4 A8 20 00 00 00 00 41 4E 84 80 00 00 00 00 ; @ä¨ ....AN„€....
00040540h: C0 D5 1B 00 00 00 00 00 40 E4 A5 80 00 00 00 00 ; ÀÕ......@ä¥€....
```

***Table 6:*** BXML (rounded) format excerpt (note the zeroes replacing pseudo-random digits)