S-101 Test dataset Notes.

The s-101 test dataset was pepared from the December 2009 edition of the S-101 product specification. This edition includes an example s-101 encoding in iso8211 format. The aim of the example dataset is to illustrate that the 8211 encoding is valid and that existing ENCs can be mapped with reasonable certainty to the new encoding. A secondary objective was to illustrate some of the new concepts being introduced wihtin S-101. For the purpose of this exercise the new features were:

- New geometry model. Line features point to a single composite curve. Area features point to a single surface, all of which are defined in the geometry section of the cell.
- Complex Attributes. All features initially translat as simple ones but a single example based on lights indexes the sectors using the modl developed for complex attributes.

Methodology.

s-57 ENCs are encoded in iso8211 as well. Iso8211 is a self-describing format where data is arranged into a simple field/subfield structure. Iso8211 files contain a descriptive header where the individual fields' data formats are defined.

Within this description I have adopted an intermediate XML encoding which simplified greatly the description of the encoding itself. This XML encoding simply turns the existing S-57 field/subfield structure into standard XML elements. An example is shown below:

```
|--<R>--Feature record
|            |--0001 -ISO/IEC 8211 Record Identifier
|            |--<1>--FRID -Feature Record Identifier field
|            |--<1>--FOID -Feature Object Identifier field
|            |--<R>--ATTF* -Feature Record Attribute field
|            |--<R>--NATF* -Feature Record National Attribute field
|            |--<R>--FFPT* -Feature Record to Feature Object Pointer field
|            |--<R>--FSPT* -Feature Record to Spatial Record Pointer field
```

will have data encoded in XML as

```
<RECORD>
    <FRID>
        <RCNM>100</RCNM>
        <RCID>17</RCID>
        <PRIM>3</PRIM>
        <GRUP>255</GRUP>
        <OBJL>302</OBJL>
        <RVER>1</RVER>
        <RUIN>1</RUIN>
    </FRID>
    <FOID>
        <AGEN>540</AGEN>
        <FIDN>0000001811</FIDN>
        <FIDS>80</FIDS>
    </FOID>
    <ATTF>
        <ATTL>18</ATTL>
        <ATVL>1</ATVL>
        <ATTL>102</ATTL>
        <ATVL>12</ATVL>
    </ATTF>
    <FSPT>
```

```
        <NAME>8226000000</NAME>
        <ORNT>1</ORNT>
        <USAG>1</USAG>
        <MASK>2</MASK>
        <NAME>820B000000</NAME>
        <ORNT>1</ORNT>
        <USAG>1</USAG>
        <MASK>2</MASK>
        <NAME>8222000000</NAME>
        <ORNT>2</ORNT>
        <USAG>1</USAG>
        <MASK>2</MASK>
        <NAME>8228000000</NAME>
        <ORNT>1</ORNT>
        <USAG>1</USAG>
        <MASK>2</MASK>
    </FSPT>
</RECORD>
```

In iso8211 terms a single XML “RECORD” above translates into an 8211 sequence of field/subfield values, each of which is preceded by a leader and catalogue.

The methodology used uses the XML as an intermediate format. S-57 ENCs are readily convertible into this format and a translation of these back into iso8211 is, though not striaghtforward, is possible with some additional specification.

S-57 ENCs have iso8211 records arranged in the following order (ENC Appendix B1) :

Table 1. S-57 Data set file

1. Data set general information record
2. Data set geographic reference record (for EN application profile)
3. Vector records
    i. Isolated nodes (SG3D)
    ii. Isolated nodes (SG2D)
    iii. Connected nodes
    iv. Edges
4. Feature records
    i. Meta features
    ii. Geo features (ordered from slave to master)
    iii. Collection features

By contrast S-101 cells have the following record order:

Table 2. Data set file

1. Data set general information record
2. Data set structure information field structure
3. Data set co-ordinate reference system record structure
4. Information records
    a. Information
5. Vector records
    a. Point
    b. Multi point
    c. Curve
    d. Composite Curve
    e. Surface
6. Feature records
    a. Meta features

b. Geo features (ordered from slave to master)
c. Aggregated features
d. Theme features

With all these factors in mind the methodology was reaosnably simple.

1. Convert an existing ENC (a sample ENC was supplied but the methodology works for a generic ENC dataset) into intermediate XML data.
2. Split the resultant XML into its component parts (as per Table 1)
3. Create new components based on the internal structure of the XML as per Table 2. This is also automatic.
4. Any further manual edits to the XML can be made (the complex attribute example is inserted in this way).
5. Re-encode into iso8211.
6. Surround with new packaging and deliver…

Conversion.

Step 1 was reaosnably simple. An existing tool to parse iso8211 was used to create the XML files required. This is based on an existing library (the GDAL iso8211 interpreter created and maintained by Frank Warmadam[1]) . This can be done for any

Step 2 required parsing the XML and splitting into the relevant fields.

Header information:
A sample dataset header and coordinate reference system record was assembled by hand and used as the header for the entire dataset. In practice very little of this data is currently recorded for ENCs but that which isn't currently held should be easy to default appropriately (e.g coordinate reference systems). The complete header is shown in the table below:

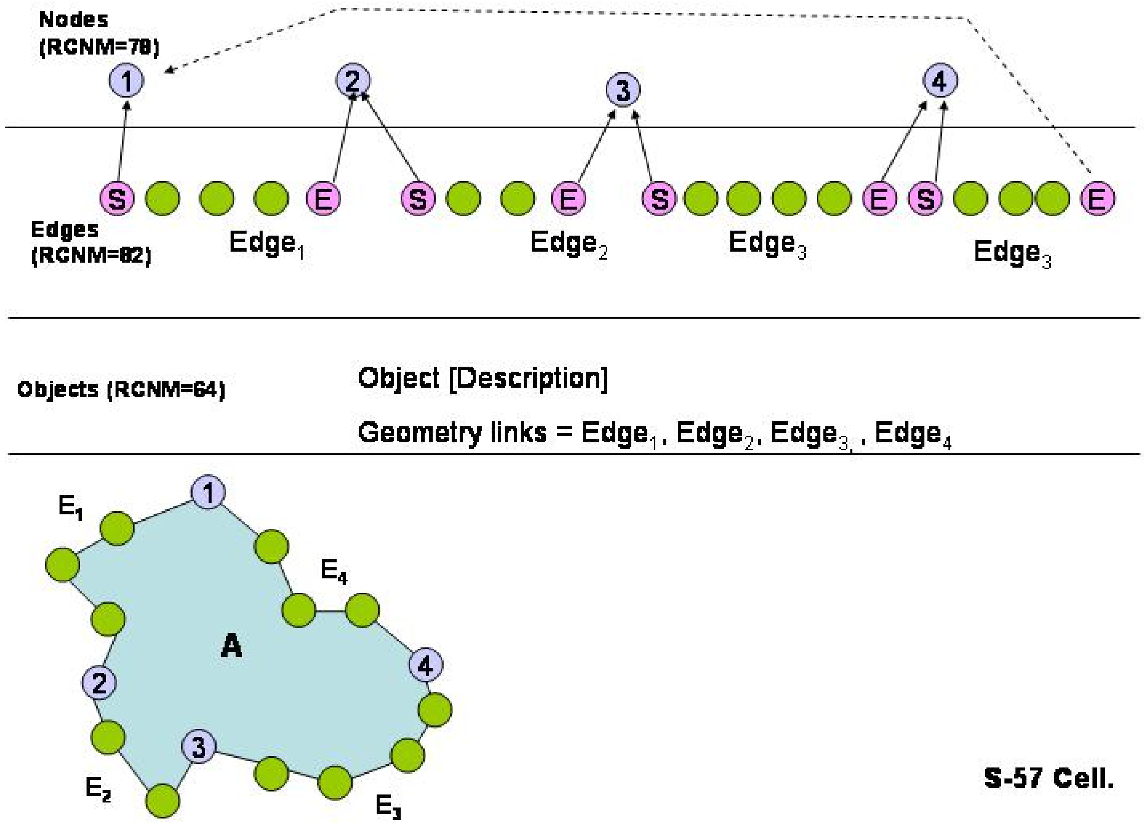
XML representation of cell content	Comments.
<RECORD>	
<DSID>	
<RCNM>10</RCNM>	
<RCID>1</RCID>	
<ENSP></ENSP>	
<ENED></ENED>	
<PRSP>1</PRSP>	
<PRED>2.0</PRED>	
<PROF>1</PROF>	
<DSNM>s101cell</DSNM>	
<DSTL>Experimental S-101 dataset</DSTL>	
<DSRD>iso8211 </DSRD>	
<DSLG>JP version 2.04EN</DSLG>	
<DSAB>test dataset made up from dummy data</DSAB>	
<DSED>2.0</DSED>	
<DSTC>1</DSTC>	
</DSID>	
<DSSI>	
<DCOX>0</DCOX>	I have not filled in any of the total fields (refinement coming later).
<DCOY>0</DCOY>	
<DCOZ>0</DCOZ>	
<CMFX>1</CMFX>	
<CMFY>1</CMFY>	
<CMFZ>1</CMFZ>	
<NOIR>0</NOIR>	

[1] In crreating this cell and investigating copnatenated data structures I have discovered that GDAL (and by implication many other libraries and tools such as FME will not parse concatenated data structures. We are attempting to rectify this with the library authors.

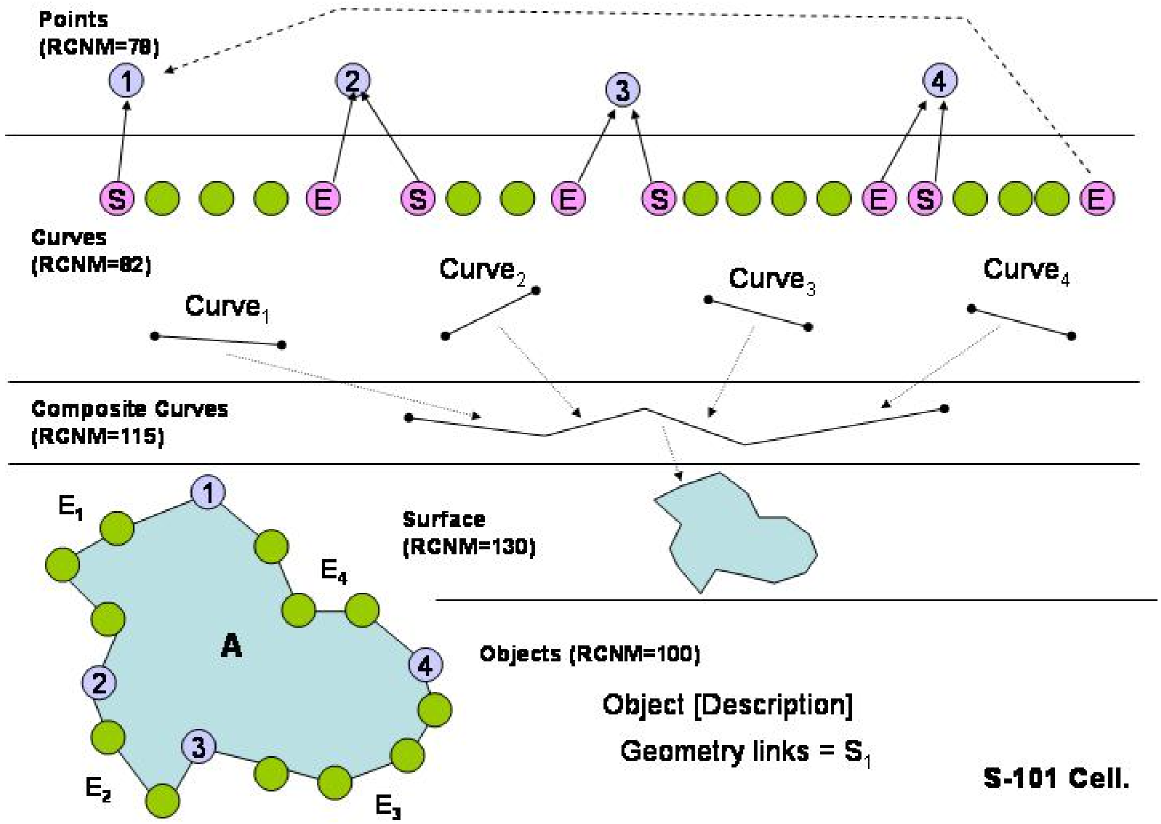
<NOPM>0</NOPM> <NOMN>0</NOMN> <NOCN>0</NOCN> <NOXN>0</NOXN> <NOSN>0</NOSN> <NOFR>0</NOFR> </DSSI> <CRSI> <RCNM>15</RCNM> <RCID>2</RCID> <NCRC>1</NCRC> </CRSI> <CRSH> <CRST>0</CRST> <CSTY>1</CSTY> <CRNM>stuff</CRNM> <CRSI></CRSI> <CRSS>1</CRSS> <SRCI>none</SRCI> </CRSH> <CSAX> <AXTY>0</AXTY> <AXUM>1</AXUM> </CSAX> </RECORD>	I’m not sure about the actual values required for these fields. They can be filled into a standard header though.

Geometry Translation.
Translation into the more complex geometric model required splitting the XML into the S-101 parts, inserting new objects as required and then assembling the links for each feature within the cell.

Translation of Object structures.
The object to spatial linkage differences between S-57 and S-101 ENCs is described in the diagram below. This shows how an area object is encoded in S-57 and S-101. S-101 gathers the individual edges together under a “surface” which is linked to by the object.


Nodes
(RCNM=78)
1
2
3
4
S
E
Edges
(RCNM=82)
Edge1
Edge2
Edge3
Edge3
Objects (RCNM=64)
Object [Description]
Geometry links = Edge1, Edge2, Edge3, , Edge4
E1
E2
E3
E4
A

S-57 Cell.



This required, therefore splitting out multi-point records from single point records as well as inserting intermediate XML records to hold composite curves and surfaces to act as reference points for the features themselves.

Feature translation is not so difficult however SCAMIN has to be taken from the ATTR fields of the object and inserted object by object. An example of a complete feature is shown below:

```
<RECORD>
    <FRID>
        <RCNM>100</RCNM>
        <RCID>137</RCID>
        <OBJC>71</OBJC>
        <RVER>0</RVER>
        <RUIN>1</RUIN>
    </FRID>
    <ATTR>
        <ATLB>147</ATLB>
        <ATIX>1</ATIX>
        <PAIX>1</PAIX>
        <ATIN>1</ATIN>
        <ATVL> 20100120</ATVL>
        <ATLB>148</ATLB>
        <ATIX>1</ATIX>
        <PAIX>1</PAIX>
        <ATIN>1</ATIN>
        <ATVL>BA 1234 </ATVL>
    </ATTR>
    <SPAS>
        <RRNM>130</RRNM>
        <RRID>115</RRID>
        <ORNT>1</ORNT>
        <SMIN>0</SMIN>
        <SMAX>1</SMAX>
```

```
        <SAUI>1</SAUI>
    </SPAS>
</RECORD>
```

which links to the following spatial record:

```
<RECORD>
    <SRID>
        <RCNM>130</RCNM>
        <RCID>115</RCID>
        <RVER>1</RVER>
        <RUIN>1</RUIN>
    </SRID>
    <RIAS>
        <RRNM>130</RRNM>
        <RRID>96</RRID>
        <ORNT>1</ORNT>
        <USAG>0</USAG>
        <RAUI>30</RAUI>
    </RIAS>
</RECORD>
```

Construction of Complex Attributes.
To encode complex attributes you need to construct a tree of the individual attribute names and values as per the example below. This is the LIGHT feature within the test data cell and in S-57 it is encoded as three features each pointing to the same point. In the S-101 form the tree below represents all the characteristics of the light

```
[00] Feature
[01]|_0_____LITSEC
[02]|        |_1____SECTR1 [1] 340
[03]|        |_1____SECTR1 [2] 330
[04]|        |_1____SECTR1 [3] 310
[05]|        |_1____COLOUR [1] 4
[06]|        |_1____COLOUR [2] 1
[07]|        |_1____COLOUR [3] 3
[08]|        |_1____VALNMR [1] 10
[09]|        |_1____VALNMR [2] 10
[10]|        |_1____VALNMR [3] 10
[11]|        |_1____SECTR2 [1] 0
[12]|        |_1____SECTR2 [2] 340
[13]|        |_1____SECTR2 [3] 330
  |
[14]|_0_____RHYLGT
[15]|        |_14___ SIGPER [1] 5
[16]|        |_14___ SIGPER [2] 3
[17]|        |_14___ SIGPER [3] 2
[18]|        |_14___ SIGGRP [1] 1
[19]|        |_14___ SIGGRP [2] 2
[20]|        |_14___ SIGGRP [3] 1
[21]|        |_14___ LITCHR [1] 2
[22]|        |_14___ LITCHR [2] 2
[23]|        |_14___ LITCHR [3] 2
  |
[24]|_0_______HEIGHT [1] 65
[25]|_0_______INFORM [1] Reported 14.1.99
[26]|_0_______CATLIT [1] 6
[27]|_0_______OBJNAM [1] Complex Light
[28]|_0_______SCAMIN Not needed… - moved into feature entry.
```

This tree is then decomposed into the following table.

Index	ATLB	ATIX	PAIX	ATIN	ATVL
1	LITSEC	1	0	1	-

2	136	1	1	1	340
3	136	2	1	1	330
4	136	3	1	1	310
5	75	1	1	1	4
6	75	2	1	1	1
7	75	3	1	1	3
8	178	1	1	1	10
9	178	2	1	1	10
10	178	3	1	1	10
11	137	1	1	1	0
12	137	2	1	1	340
13	137	3	1	1	330
14	RHYLGT	1	0	1	-
15	142	1	14	1	5
16	142	2	14	1	3
17	142	3	14	1	2
18	141	1	14	1	1
19	141	2	14	1	2
20	141	3	14	1	1
21	107	1	14	1	2
22	107	2	14	1	2
23	107	3	14	1	2
24	HEIGHT (95)	1	0	1	65
25	INFORM (102)	1	0	1	Reported 14.11.99
26	CATLIT (37)	1	0	1	6
27	OBJNAM (116)	1	0	1	Complex Light

This table then forms the basis of the XML which is encoded in the cell. The XML for this particular light is listed with the rest of the cell at the end of this document.

Compilation of the complete cell.

The compilation of the complete cell is composed of parsing and 8211 encoding of the input XML files which have been compiled according to the revised S-101 structure

The 8211 encoding uses the following specification (this is my specification of the information required for an iso8211 header.. The fields are as follows.

<XXXX – XXXX=8211 field name
"YYYY" – Long name of 8211 field name
x1!x2!x3! – Sequence of subfields within the field.
(b1,b2,b3) - encoding specifications for each subfield.
N - field code, Mixed = 6, specified in iso8211.
- a commented field. This is a line which I have altered from the original encoding.

Perhaps the single biggest change in the 821 encoding is the use of concatenated fields. These are specified with a "\\" in the subfield array and are present in a number of the S-101 encodings. They represent two structures joined together within a single 8211 field and are generally used to represent a number of data items joined to an arrray of values, e.g a single vertical datum id followed by a number of coordinates all of which are based on that vertical datum id. Without the use of concatenated fields the vertical datum id would need to be replicated for every coordinate pair. Although the use of concatenated fields is perfectly valid the current version of the cell does not use them, instead I have replaced it with two separate 8211 fields (I will implement concatenanted fields in a later edition of the cell once I have tools which can produce and parse them!).

Additonally I have altered some of the encoding specifications to make it easier to write data. These will be changed back at a later date.

```
<0000 "NULL" 0001DSIDDSID0001CRID0001PRIDPRIDATTRPRIDINASPRIDC2DIPRIDC3DI NULL 0
<0001 "RecordID" NULL (b12) 5
#<DSID "Data Set Identification"
RCNM!RCID!ENSP!ENED!PRSP!PRED!PROF!DSNM!DSTL!DSRD!DSLG!DSAB!DSED\\*DSTC
(b11,b14,7A,A(8),3A,b11) 6
<DSID "Data Set Identification"
RCNM!RCID!ENSP!ENED!PRSP!PRED!PROF!DSNM!DSTL!DSRD!DSLG!DSAB!DSED!DSTC
(b11,b14,7A,A(8),3A,b11) 6
<DSSI "Data Set Structure Information"
DCOX!DCOY!DCOZ!CMFX!CMFY!CMFZ!NOIR!NOPM!NOMN!NOCN!NOXN!NOSN!NOFR (3b48,10b14) 6
<CRID "Coordinate Reference System Record Identifier" RCNM!RCID!NCRC (b11,b14,b11) 1
<CRSH "Coordinate Reference System Header" CRST!CSTY!CRNM!CRSI!CRSS!SRCI (2b11,2A,b11,A) 6
<CSAX "Coordinate System Axes" *AXTY!AXUM (2b11) 1
<PROJ "Projection" PROM!PRP1!PRP2!PRP3!PRP4!PRP5!FEAS!FNOR! (b11,7b48) 6
<GDAT "Geodetic Datum" DTNM!ELNM!ESMA!ESPT!ESPM!CMNM!CMGL! (2A,b48,b11,b48,A,b48) 6
<VDAT "Vertical Datum" DTIX!DTNM!DTID!DTSR!SCRI! (b11,2A,b11,A) 6
#<IRID "Information Type Record Identifier" RCNM!RCID!OBJC!RVER!RUIN (b11,b14,2b12,b11) 1
<INAS "Information Association" *RRNM!RRID!IROL!IUIN (b11,b14,b12,b11) 1
<ATTR "Attribute" *ATLB!ATIX!PAIX!ATIN!ATVL (3b12,b11,A) 6
<COCC "Coordinate Control" COUI!COIX!NCOR (b11,2b12) 1
<C2DI "2-D Integer Coordinate" *YCOO!XCOO (2b24) 1
<C2DF "2-D Floating Point Coordinate" *YCOO!XCOO (2b48) 2
#<C3DF "3-D Floating Point Coordinate" VDID\\*YCOO!XCOO!ZCOO (b11,3b48) 6
#<C3DI "3-D Integer Coordinate" VDID\\*YCOO!XCOO!ZCOO (b11,3b24) 1
<VDID "3-D Vertical Datum id" VDID (b11) 1
<C3DF "3-D Floating Point Coordinate" *YCOO!XCOO!ZCOO (3b48) 1
<C3DI "Modified 3-D Int" *YCOO!XCOO!ZCOO (3b24) 1
<PRID "Point Record Identifier" RCNM!RCID!RVER!RUIN (b11,b14,b12,b11) 1
<MRID "Multi Point Record Identifier" RCNM!RCID!RVER!RUIN (b11,b14,b12,b11) 1
<CRID "Curve Record Identifier" RCNM!RCID!RVER!RUIN (b11,b14,b12,b11) 1
<PTAS "Point Association" *RRNM!RRID!TOPI (b11,b14,b11) 1
<SECC "Segment Control" SEUI!SEIX!NSEG (b11,2b12) 1
#<SECC "Segment Header" INTP (b11) 1
<CCID "Composite Curve Record Identifier" RCNM!RCID!RVER!RUIN (b11,b14,b12,b11) 1
```

```
<CCOC "Curve Component Control" CCUI!CCIX!NCCO (b11,2b12) 1
<CUCO "Curve Component" *RRNM!RRID!ORNT (b11,b14,b11) 1
<SRID "Surface Record Identifier" RCNM!RCID!RVER!RUIN (b11,b14,b12,b11) 1
<RIAS "Ring Association" RRNM!RRID!ORNT!USAG!RAUI (b11,b14,3b11) 1
<FRID "Feature Type Record Identifier" RCNM!RCID!OBJC!RVER!RUIN (b11,b14,2b12,b11) 1
<FOID "Feature Object Identifier" AGEN!FIDN!FIDS (b12,b14,b12) 1
<SPAS "Spatial Association" *RRNM!RRID!ORNT!SMIN!SMAX!SAUI! (b11,b14,b11,2b14,b11) 1
<FEAS "Feature Association" *RRNM!RRID!ASCD!RLCD!APUI (b11,b14,2b12,b11) 1
<THAS "Theme Association" *RRNM!RRID!TAUI (b11,b14,b11) 1
<MASK "Masked Spatial Record" *RRNM!RRID!MUIN (b11,b14,b11) 1
```

The encoding itself is composed of a valid 8211 header (made up from the names/titles above) complete with catalogue entries derived from the XML and field width specifiers. The 8211 is produced from the XML in a reversible process. I have inserted all the XML as a separate document.

Caveats and health warnings:!!!

This cell is very very much a first draft and I have drawn attention to any shortcomings in the process of translation and encoding. There may well be issues in the actual values for certain fields. A thorough review of the cell would pick these up and would allow refinement. However, the work done to date suggests that a wholly deterministic process exists for translating S-57 encoded ENC data into S-101 encoded data (whether in iso8211 or not).