

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29



EPCglobal Tag Data Translation (TDT) 1.0  
Ratified Standard Specification  
January 21, 2006

**Copyright notice**

© 2006, EPCglobal Inc.

All rights reserved. Unauthorized reproduction, modification, and/or use of this document is not permitted. Requests for permission to reproduce should be addressed to [epcglobal@epcglobalinc.org](mailto:epcglobal@epcglobalinc.org).

**Disclaimer**

EPCglobal Inc.<sup>TM</sup> is providing this document as a service to interested industries. This document was developed through a consensus process of interested parties. Although efforts have been to assure that the document is correct, reliable, and technically accurate, EPCglobal Inc. makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS DOCUMENT IS CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR WILL BE SUITABLE FOR ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of this document is with the understanding that EPCglobal Inc. has no liability for any claim to the contrary, or for any damage or loss of any kind or nature.

## 30 **Table of Contents**

31	Terminology.....	5
32	Status of this document.....	5
33	1. Introduction.....	6
34	1.1. Overview .....	6
35	1.2. Tag Data Translation Charter .....	7
36	1.3. Tag Data Translation Concept.....	7
37	1.4. Role within the EPC Network Architecture .....	9
38	1.5. Tag Data Translation Process.....	12
39	1.6. Expressing different representations of EPC.....	15
40	1.6.1. Patterns (Regular Expressions) .....	15
41	1.6.2. Grammar (Augmented Backus-Naur Form [ABNF]) .....	15
42	1.6.3. Rules for obtaining additional fields .....	15
43	1.7. Translation Process Steps .....	16
44	2. Tag Data Standards .....	16
45	2.1. Overview .....	16
46	2.2. Many Schemes, 4 Levels within each scheme and multiple options for each level.....	18
47	3. TDT Markup and Logical Process.....	20
48	3.1. TDT Master Index file.....	20
49	3.2. TDT Markup.....	21
50	3.3. Definition of Formats via Regular Expression Patterns and ABNF Grammar .....	23
51	3.4. Determination of the inbound representation .....	23
52	3.5. Specification of the outbound representation .....	24
53	3.6. Specifying supplied parameter values .....	24
54	3.7. Validation of values for fields and fields derived via rules .....	26
55	3.8. Restricting and checking decimal ranges for values of fields .....	26
56	3.9. Restricting and checking character ranges for values of fields .....	27
57	3.10. Padding of fields.....	28
58	3.11. Compaction and Compression of fields.....	31
59	3.12. Names of fields used within the TDSv1.1r1.27 schemes .....	31
60	3.13. Rules and Derived Fields.....	32

61	3.14. Core Functions.....	34
62	4. TDT Markup - Elements and Attributes .....	38
63	4.1. Root Element .....	38
64	4.1.1. Attributes.....	38
65	4.1.2. Elements.....	38
66	4.2. Scheme Element .....	38
67	4.2.1. Attributes.....	38
68	4.2.2. Elements.....	39
69	4.3. Level Element.....	39
70	4.3.1. Attributes.....	39
71	4.3.2. Elements.....	40
72	4.4. Option Element.....	40
73	4.4.1. Attributes.....	40
74	4.4.2.....	40
75	4.4.3. Elements.....	41
76	4.5. Field Element.....	41
77	4.6. Attributes .....	41
78	4.7. Rule Element .....	42
79	4.7.1. Attributes.....	42
80	5. Translation Process .....	44
81	5.1. Tag Data Translation Software - Reference Implementation.....	46
82	6. Application Programming Interface .....	46
83	6.1. Client API.....	47
84	6.2. Maintenance API .....	48
85	7. TDT Schema Definition.....	49
86	8. TDT Markup V1.0 .....	55
87	8.1. GID-96 General Identifier .....	55
88	8.2. SGTIN-64 Serialized Global Trade Item Number for 64-bit EPCs .....	56
89	8.3. SGTIN-96 Serialized Global Trade Item Number for 96-bit EPCs .....	61
90	8.4. SSCC-64 Serial Shipping Container Code for 64-bit EPCs.....	66
91	8.5. SSCC-96 Serial Shipping Container Code for 96-bit EPCs.....	70
92	8.6. SGLN-64 Serialized Global Location Number for 64-bit EPCs.....	73
93	8.7. SGLN-96 Serialized Global Location Number for 96-bit EPCs.....	78

94 8.8. GRAI-64 Global Returnable Asset Identifier for 64-bit EPCs..... 82

95 8.9. GRAI-96 Global Returnable Asset Identifier for 96-bit EPCs..... 87

96 8.10. GIAI-64 Global Individual Asset Identifier for 64-bit EPCs ..... 91

97 8.11. GIAI-96 Global Individual Asset Identifier for 96-bit EPCs ..... 94

98 8.12. USDOD-64 DEPARTMENT OF DEFENCE for 64-bit EPCs..... 98

99 8.13. USDOD-96 DEPARTMENT OF DEFENCE for 96-bit EPCs..... 99

100 9. Glossary (non-normative)..... 100

101 10. References..... 106

102

## 103 **Terminology**

104 Within this specification, the terms SHALL, SHALL NOT, SHOULD, SHOULD NOT,  
105 MAY, NEED NOT, CAN, and CANNOT are to be interpreted as specified in Annex G of  
106 the ISO/IEC Directives, Part 2, 2001, 4th edition [ISODir2]. When used in this way,  
107 these terms will always be shown in ALL CAPS; when these words appear in ordinary  
108 typeface they are intended to have their ordinary English meaning.

109 The `Courier` font is used to indicate the names of XML elements and attributes and  
110 names of variable fields within the Tag Data Translation markup.

111 All sections of this document are normative, except where explicitly noted as non-  
112 normative.

## 113 **Status of this document**

114 This section describes the status of this document at the time of its publication. Other  
115 documents may supersede this document. The latest status of this document series is  
116 maintained at EPCglobal. This document is the version called the Ratified Standard  
117 Specification and was Ratified by the EPCglobal Board of Governors..

118 Comments on this document should be sent to the EPCglobal Software Action Tag Data  
119 Translation Working Group mailing list,  
120 [mailto:sag\\_tagdatatranslation@epclinklist.epcglobalinc.org](mailto:sag_tagdatatranslation@epclinklist.epcglobalinc.org).

# 121 1. Introduction

## 122 1.1. Overview

123 The Electronic Product Code (EPC) is a globally unique identifier that is designed to  
124 allow the automatic identification of objects anywhere.

125 The EPC Tag Data Standards (TDS) specification indicates how legacy coding systems  
126 such as the GS1 (formerly EAN.UCC) family of codes (GTIN, GLN, SSCC, GRAI,  
127 GIAI) should be embedded within the Electronic Product Code (EPC).

128 In the future, other industry sectors such as automotive, aviation and defence may call for  
129 the embedding of their existing numbering schemes directly within the EPC, especially  
130 for applications where they deem that it is impractical to map the original general-  
131 purpose General Identifier (GID) scheme of EPC to the corresponding identifier in their  
132 legacy coding system via a simple network lookup.

133 The EPC Tag Data Standards (TDS) specification also describes in terms of human-  
134 readable encoding and decoding rules for each coding scheme, how to translate between  
135 three representations of the electronic product code (EPC), namely the binary format and  
136 two formats of uniform resource identifiers (URI), one for tag-encoding and another for  
137 pure identity.

138 *The binary format is used to store the EPC identifier in the memory of the RFID tag.*  
139 *EPC Tag Data Standards v1.1 defines binary formats consisting of either 64 bits or 96*  
140 *bits. The binary format consists of a header (which indicates the coding scheme and*  
141 *version - usually the first 8 bits, although a 2-bit header is defined for SGTIN-64), a fast*  
142 *filter value (which can be used for distinguishing between different packaging levels), as*  
143 *well as fields indicating the company responsible for the object, the object class and a*  
144 *unique serial number.*

145 *The URI (or strictly speaking URN) representations are intended for communicating and*  
146 *storing EPCs in information systems, databases and applications, in order to insulate*  
147 *them from knowledge about the physical nature of the tag, so that although 64 bit tags*  
148 *may differ from 96 bit tags in the choice of literal binary header values and the number*  
149 *of bits allocated to each element or field within the EPC, the URN format does not*  
150 *require the information systems to know about these details; the URN can be just a pure*  
151 *identifier.*

152 *The tag-encoding URI provides a 1-1 mapping with the binary number recorded in the*  
153 *physical tag and as such indicates the bit-length of the tag and may also include an*  
154 *additional field (usually 3 bits) which is reserved for fast filtering purposes, e.g. to*  
155 *distinguish between various packaging levels for trade items. The tag-encoding URI is*  
156 *therefore intended for low-level applications which need to write EPCs to tags or*  
157 *physically sort items based on packaging level.*

158 *The pure-identity URI format isolates the application software from details of the bit-*  
159 *length of the tags or any fast filtering values, so that tags of different bit-lengths which*  
160 *code for the same unique object will result in an identical pure-identity URI, even though*  
161 *their tag-encoding URIs and binary representations may differ. This means that when a*

162 *manufacturer switches from using 64-bit tags to 96-bit tags or longer for tagging a*  
163 *particular product, the pure-identity URI representation of the EPC will appear the same*  
164 *(except for different serial numbers for different instances of the product).*

165 This EPC Tag Data Translation (TDT) specification is concerned with a machine-  
166 readable version of the EPC Tag Data Standards specification. The machine-readable  
167 version can be readily used for validating EPC formats as well as translating between the  
168 different levels of representation in a consistent way. This specification describes how to  
169 interpret the machine-readable version. It contains details of the structure and elements  
170 of the machine-readable markup files and provides guidance on how it might be used in  
171 automatic translation or validation software, whether standalone or embedded in other  
172 systems.

## 173 **1.2. Tag Data Translation Charter**

174 The three objectives in the charter of the Tag Data Translation working group are:

- 175 • To develop the necessary specifications to express the current TDS encoding and  
176 decoding rules in an unambiguous machine-readable format; this will allow any  
177 component in the EPC Network technology stack to automatically translate between  
178 the binary and tag-encoding URI and pure-identity URI formats of the EPC as  
179 appropriate. The motivation is to allow components flexibility in how they receive or  
180 transmit EPCs, to reduce potential 'impedance mismatches' at interfaces in the EPC  
181 Network technology stack. Reference implementations of software that demonstrate  
182 these capabilities will also be developed.
- 183 • To provide documentation of the TDS encodings in such a way that the current prose  
184 based documentation can be supplemented by the more structured machine-readable  
185 formats.
- 186 • To ensure that automated tag data translation processes can continue to function and  
187 also handle additional numbering schemes, which might be embedded within the EPC  
188 in the future. By aiming for a future-proof mechanism which allows for smooth  
189 upgrading to handle longer tags (e.g. 256 bits) and the incorporation of additional  
190 encoding/decoding rules for other coding systems, we expect to substantially reduce  
191 the marginal cost of redeveloping and upgrading software as the industry domains  
192 covered by the EPC expand in the future. We envisage that data which specifies the  
193 new rules for additional coding schemes will be readily available for download in  
194 much the same way as current anti-virus software can keep itself up to date by  
195 periodically downloading new definition files from an authoritative source.

## 196 **1.3. Tag Data Translation Concept**

197 The Tag Data Translation process translates one representation of EPC into another  
198 representation, within a particular coding scheme. For example, it could translate from  
199 the binary format for a GTIN on a 64-bit tag to a pure-identity URI representation of the  
200 same identifier, although it could not translate a SSCC into a SGTIN or vice versa.

201 The Tag Data Translation concept is illustrated in Figure 1.

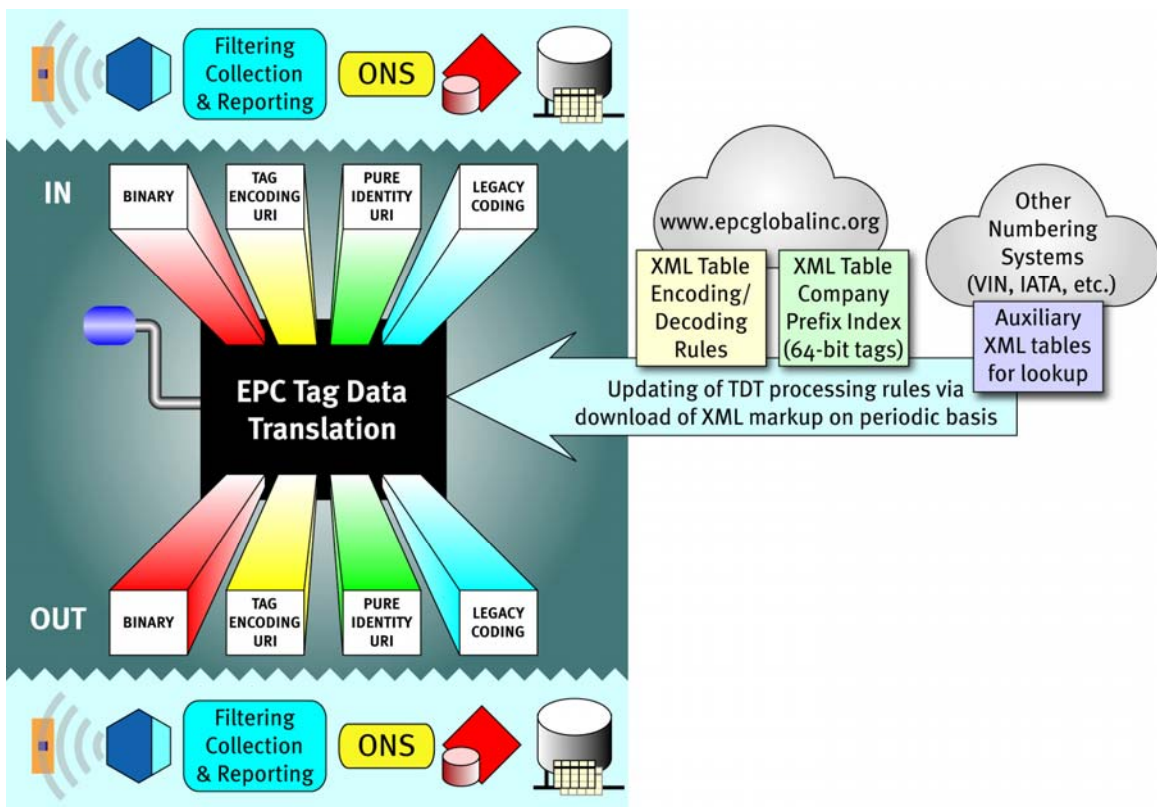


Figure 1 - Tag Data Translation - Concept

202  
203  
204

205 Tag Data Translation capabilities may be implemented at any level of the EPC Network  
 206 stack, from readers, through filtering middleware, as a pre-resolver to the Object Name  
 207 Service (ONS), as well as by applications and networked databases complying with the  
 208 EPCIS interface. Tag Data Translation converts between different levels of representation  
 209 of the EPC and may make use of external tables, such as the Company Prefix Index  
 210 lookup table for 64-bit tags. It is envisaged that Tag Data Translation software will be  
 211 able to keep itself up-to-date by periodically checking for and downloading TDT markup  
 212 files, although a continuous network connection should not be required for performing  
 213 translations or validations, since the TDT markup files and any auxiliary tables can be  
 214 cached between periodic checks; in this way a generic translation mechanism can be  
 215 extensible to further coding schemes or variations for longer tag lengths, which may be  
 216 introduced in the future.

217 *Although the TDT markup files are made available in XML format, this does not impose a*  
 218 *requirement for all levels of the EPC Network technology stack to implement XML*  
 219 *parsers, nor to obtain the TDT markup files directly from the authoritative source, such*  
 220 *as EPCglobal and possibly other numbering authorities who may develop TDT markup*  
 221 *files for other coding schemes / other industry sectors. For example, the manufacturer of*  
 222 *an RFID reader may regularly check for and obtain the current TDT markup files, then*  
 223 *use data binding software to convert these into hierarchical software data objects, which*  
 224 *could be saved more compactly as serialized objects accessible from the particular*  
 225 *programming language in which their reader software/firmware is written. The reader*



226 *manufacturer could make these serialized objects available for download to owners of*  
227 *their products – or bundle them with firmware updates, thus eliminating the need for*  
228 *either embedded or real-time parsing of the TDT markup files in their original XML*  
229 *format at the reader level.*

#### 230 **1.4. Role within the EPC Network Architecture**

231 In the EPC Network Architecture as depicted in Figure 2 below, the green bars denote  
232 interfaces governed by EPCglobal standards, while the blue boxes denote roles played by  
233 hardware and/or software components of the system.

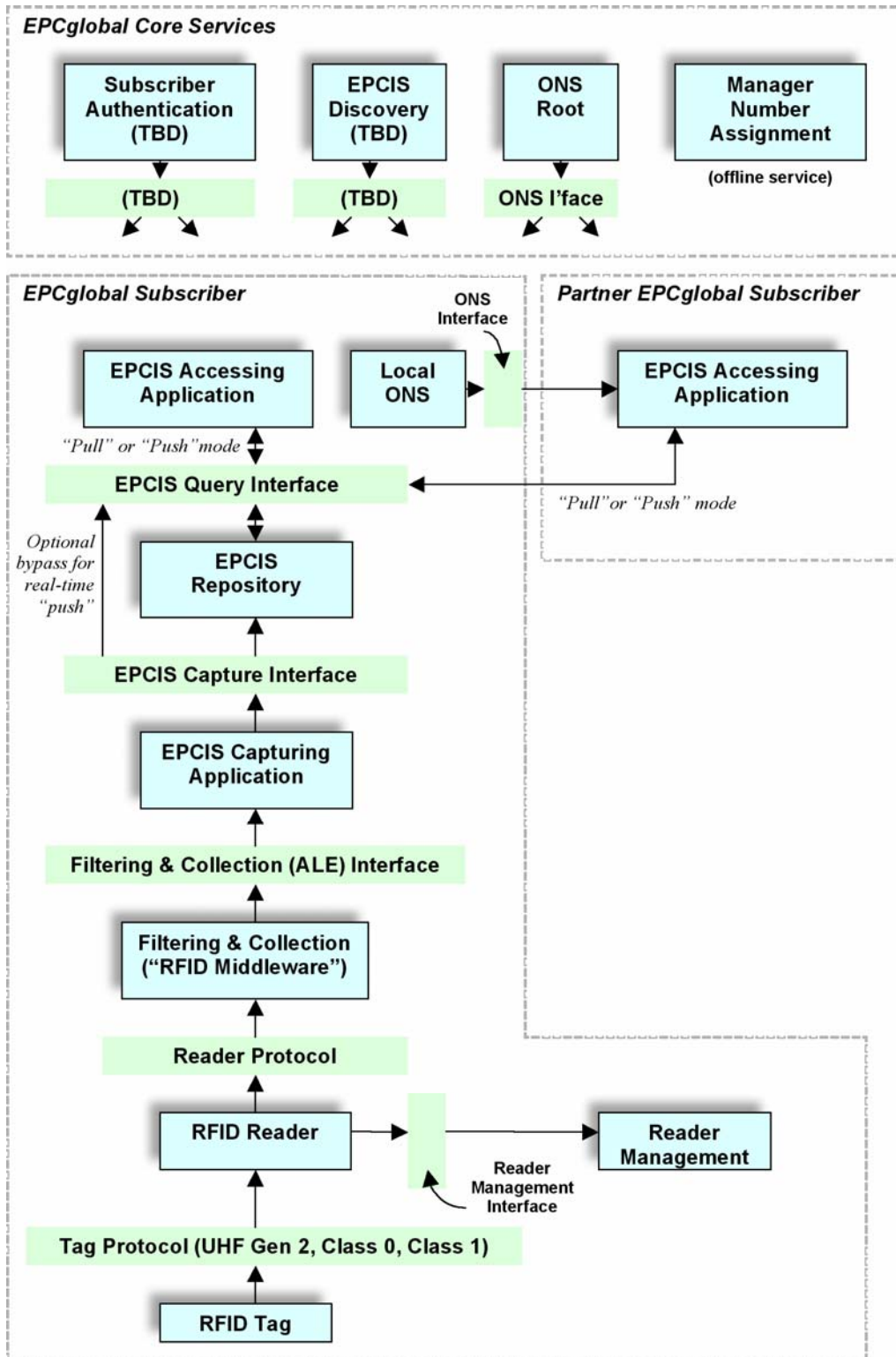


Figure 2 - EPC Network Architecture diagram

234  
235  
236

237 Table 1 describes the key elements of the EPC Network and the potential usages for the  
 238 Tag Data Translation process for encoding and decoding tag data standards.

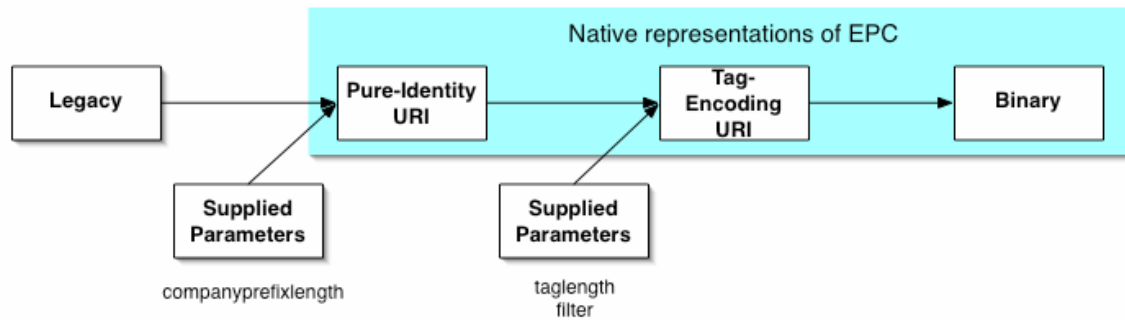
EPC Network Standards	Description	TDT Role	Potential TDT Usage
Reader Protocol	Defines the control and delivery of raw tag reads from Readers to F&C Middleware	Yes	Conversion upon 'impedance mismatch' of EPC representation
Application Level Events (ALE) Filtering & Collection	API for software that filters and collects raw tag reads, over time intervals delimited by event cycles as defined by applications such as the EPCIS Capturing Application	Yes	Conversion of other EPC representations into URI format for reports  Assistance with converting declarative URI patterns into combinations of bit-mask
EPCIS Capturing Application	Software that supervises the operation of the lower EPC network elements and coordinates with enterprise level business events	Yes	Conversion upon 'impedance mismatch' of EPC representation
ONS	ONS is a network service layered over the existing Domain Name System that is used to lookup authoritative pointers to EPCIS-enabled Repositories and other EPC-related information services, given an EPC Manager Number or full Electronic Product Code	No	TDT could output a format which is the hostname for DNS type 35 lookup, in order to perform an ONS query  ONS/TDS work groups would need to define this for all coding schemes specified in TDS
EPCIS Service Repository	Networked database or information system providing query/update access to EPC-related data	No	In underlying databases, EPCs might be stored in other formats (e.g. GTIN+serial, separately – or hexadecimal). TDT can convert these to URI formats
EPCIS Enabled Application	Application software responsible for carrying out overall enterprise business processes, such as warehouse management, shipping and receiving	No	Conversion upon 'impedance mismatch' of EPC representation
Trading Partner Application	Trading Partner software that performs the role of an EPCIS Accessing Application.	No	Conversion upon 'impedance mismatch' of EPC representation

239

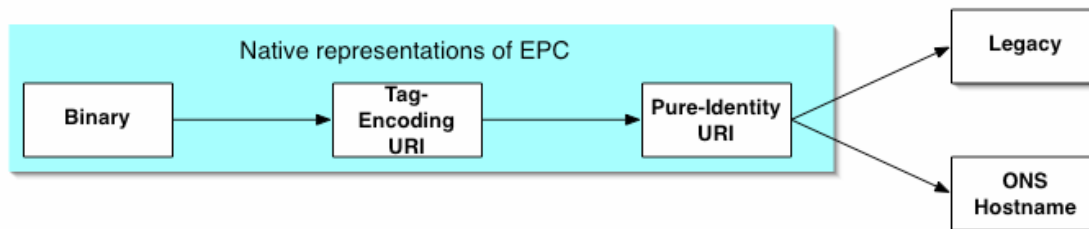




## ENCODING



## DECODING



285

286 *Figure 5 - Encoding and Decoding between different representations of an EPC. Note*  
287 *that when encoding, additional parameters need to be supplied.*

288

289 In Figure 5 above, there are actually two distinct groups of supplied parameters – those  
290 such as `companyprefixlength` which are required for parsing the input value when  
291 it is a legacy code – and others such as `filter` and `taglength`, which are required to  
292 format the output for certain levels of representation, such as binary or tag-encoding URI.  
293 In order to assist Tag Data Translation software in checking that all the required  
294 information has been supplied to perform a translation, the `<level>` elements of the  
295 Tag Data Translation markup files may contain the attribute  
296 `requiredParsingParameters` to indicate which parameters are required for  
297 parsing input values from that level and `requiredFormattingParameters` to  
298 indicate which parameters are required for formatting the output at that outbound  
299 representation level. Further details on these attributes appear in Chapter 4, which  
300 describes the TDT markup files.

301 A desirable feature of a Tag Data Translation process is the ability to automatically detect  
302 both the coding scheme and the inbound representation of the input value. This is  
303 particularly important when multiple tags are being read – when potentially several  
304 different coding schemes could all be used together and read simultaneously.

305 *For example, a shipment arriving on a pallet may consist of a number of cases tagged*  
306 *with SGTIN identifiers and a returnable pallet identified by a GRAI identifier but also*  
307 *carrying an SSCC identifier to identify the shipment as a whole. If a portal reader at a*

308 *dock door simply returns a number of binary EPCs, it is helpful to have translation*  
309 *software which can automatically detect which binary values correspond to which coding*  
310 *scheme, rather than requiring that the coding scheme and inbound representation are*  
311 *specified in addition to the input value.*

## 312 **1.6. Expressing different representations of EPC**

### 313 **1.6.1. Patterns (Regular Expressions)**

314 Given an input value, regular expression patterns may be used to match and extract  
315 groups of characters, digits or bits from the input value, in order that their values may  
316 later be used for constructing the output value in the desired outbound representation,  
317 after suitable manipulation, such as binary – decimal conversion, padding etc. We refer  
318 to these variable parts as 'fields'. Examples of fields include the Company Prefix (which  
319 usually identifies the manufacturer), the Serial Number, Fast Filter value etc.

### 320 **1.6.2. Grammar (Augmented Backus-Naur Form [ABNF])**

321 An Augmented Backus-Naur Form (ABNF) grammar may be used to express how the  
322 output is reassembled from a sequence of literal values such as URN prefixes and fixed  
323 binary headers with the variable components, i.e. the values of the various fields. For the  
324 grammar attributes of the TDT markup files, in accordance with the ABNF grammar  
325 conventions, fixed literal strings SHALL be single-quoted, whereas unquoted strings  
326 SHALL indicate that the value of the field named by the unquoted string SHOULD BE  
327 inserted in place of the unquoted string.

### 328 **1.6.3. Rules for obtaining additional fields**

329 However, not all fields that are required for formatting the output value are obtained  
330 directly from pattern-matching of the inbound representation. Sometimes additional  
331 fields are required to be known. For example, when translating a SGTIN-64 from binary  
332 to legacy codes, it will be possible to extract a Company Prefix Index, Item Reference  
333 and Serial Number from pattern-matching on the binary input – but the outbound  
334 representation needs other fields such as Company Prefix, Check Digit, Indicator Digit,  
335 which SHOULD be derived from the fields extracted from the inbound representation.  
336 For this reason, the TDT markup files also include sequences of rules, mainly within the  
337 legacy codes and binary levels. The rules express how such additional fields may be  
338 calculated or obtained via functions operating on fields whose values are already known.

339 Furthermore, there are some fields that cannot even be derived from fields whose values  
340 are already known and which SHALL therefore be specified independently as supplied  
341 parameters. For example, when translating a legacy GTIN value together with a serial  
342 number into the binary representation, it is necessary to specify independently which  
343 length of tag to use (whether 64-bit or 96-bit) and also the fast filter value to be used.  
344 Such supplied parameters would be specified in addition to specifying the input value and  
345 the desired outbound representation. As illustrated in Figure 5 , additional parameters  
346 SHOULD be supplied together with the input value when performing encoding. For  
347 decoding, it is generally not necessary to supply any additional parameters.

## 348 **1.7. Translation Process Steps**

349 There are five fundamental steps to a translation:

- 350 1. Use of the prefix matches and regular expression patterns to automatically detect  
351 the inbound representation and coding scheme of the supplied input value
- 352 2. Using the regular expression pattern to extract values of fields from the input  
353 value
- 354 3. Manipulation, (string manipulation, binary – decimal/alphanumeric conversion,  
355 padding etc.) of values of those fields in order to translate from the inbound  
356 representation to the outbound representation
- 357 4. Using the rules to calculate any additional fields required for the output
- 358 5. Using the ABNF grammar to format the required fields in the appropriate output  
359 representation

360 Note that the `prefixMatch` attribute in the TDT markup files is provided to allow  
361 optimization of software implementations to perform auto-detection of input  
362 representation more efficiently. Where multiple `option` elements are specified within a  
363 particular `level` element, each will generally have a `pattern` attribute with a subtly  
364 different regular expression as its value. The `prefixMatch` attribute of the enclosing  
365 `level` element expresses an initial prefix of these patterns which is common to all of the  
366 nested options. Optimized software need not test each nested option for a pattern match  
367 if the value of the `prefixMatch` attribute fails to match at the start of the input value.  
368 Only for those levels where the `prefixMatch` attribute matches at the start of the string  
369 should the patterns of the nested options be considered for matching.

370 Note that in the TDT markup files, the `prefixMatch` attribute SHALL be expressed as  
371 a substring to match at the beginning of the input value. The `prefixMatch` attribute  
372 SHOULD NOT be expressed in the TDT markup files as a regular expression value,  
373 since a simple string match should suffice. Software implementations MAY convert the  
374 `prefixMatch` attribute string value into a regular expression, if preferred, for example by  
375 prefixing with a leading caret `['^']` symbol (to require a match at the start of the string)  
376 and by escaping certain characters as required, e.g. escaping the dot character as `'\.'` or  
377 `'\\.'`.

## 378 **2. Tag Data Standards**

### 379 **2.1. Overview**

380 In the EPC Tag Data Standards specification, the intention is that the Electronic Product  
381 Code (EPC<sup>®</sup>) may be stored in binary representation on the physical tag, but that it is  
382 communicated within and between information systems in URI format, of which two are  
383 defined: the tag-encoding URI which contains all of the same information present in the  
384 binary representation of the physical tag and a pure-identity URI to be used by higher-  
385 level applications which are not concerned with the nature of the physical tag in which



386 the EPC was encoded. We therefore have three representations of the Electronic Product  
 387 Code, namely binary, tag-encoding URI and pure-identity URI.

388 Furthermore, the EPC Tag Data Standards specification (v1.1) describes how a number of  
 389 the GS1 (formerly EAN.UCC) coding schemes (GTIN, SSCC, GLN, GRAI and GIAI)  
 390 should be embedded within the EPC for 64-bit and 96-bit tags. The Electronic Product  
 391 Code (EPC) is intended to enable unique identification of any object anywhere  
 392 automatically. While some of the existing coding schemes (SSCC, GRAI and GIAI) are  
 393 already fully serialised, others (GTIN, GLN) are extended with a serial number for use  
 394 with the EPC and denoted SGTIN, SGLN.

395 Although technically the serialised GS1 codes are not themselves a representation of the  
 396 EPC, they can be encoded into- and decoded from the three representations of EPC, as  
 397 described in the EPC Tag Data Standards specification – so for this reason we consider  
 398 four representation levels for a EPC Tag Data Translation process as illustrated in Table  
 399 2.

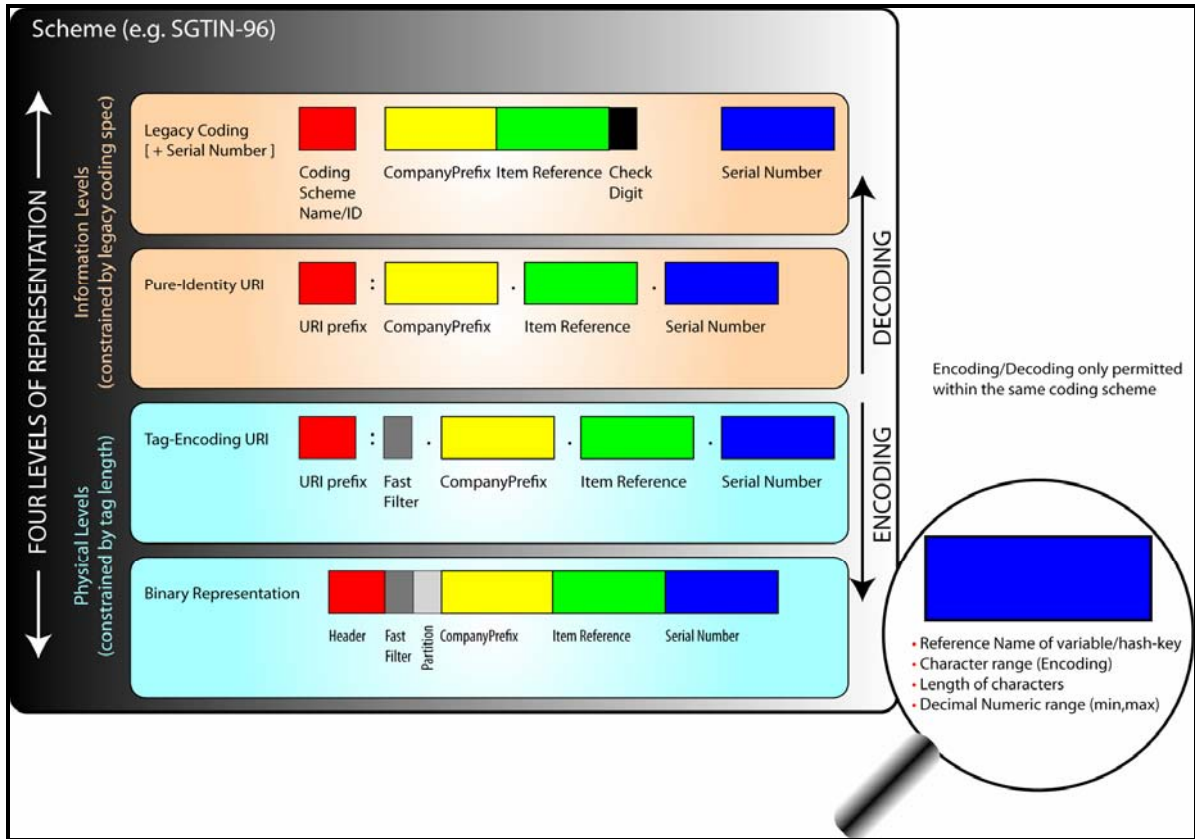
400

E N C O D E  ↓	5	Hostname for DNS type 35 query in order to perform an ONS lookup	Output-only format	↑ D E C O D E
	4	Serialised legacy coding (SGTIN,SSCC,SGLN,GRAI,GIAI)	Constrained by specifications of legacy coding schemes Does not express tag length, filter value	
	3	Pure-identity URI format of EPC		
	2	Tag-encoding URI format of EPC	Constrained by number of bits available in physical tag. Expresses tag length, filter value	
	1	Binary representation of EPC		

401 *Table 2 - Four Levels involved in the Translation Process*

402 As Table 2 indicates, the four 'levels' involved in the translation process are not  
 403 completely equivalent. There is a one-to-one mapping between the pair of levels  
 404 numbered 1 and 2 (binary and tag-encoding URI) and between the pair of levels  
 405 numbered 3 and 4 (pure-identity URI and serialized legacy code). The levels 3 and 4  
 406 lack the information present in levels 1 and 2 about tag length and fast filtering value.  
 407 This is illustrated in more detail in Figure 6 below. Note that for convenience, it may  
 408 prove useful to include a fifth 'level' of representation, corresponding to the hostname for  
 409 which a DNS Type 35 (NAPTR) query should be performed in order to effect an ONS  
 410 lookup. This is not strictly an equivalent level of representation of EPC, since ONS v1.0  
 411 does not currently provide serial-level pointers for all coding schemes. It is therefore an  
 412 output-only format and not a valid input format for encoding purposes. For this reason,  
 413 only an ABNF grammar would be defined for formatting the output in the ONS hostname  
 414 representation – and no regular expression would be defined for parsing the ONS  
 415 hostname representation as input. i.e. in the TDT markup files, the pattern attribute

416 SHALL always be absent from the level1 element representing the ONS hostname  
 417 format. This SHALL indicate to translation software that any auto-detection of the  
 418 inbound representation SHALL NOT consider the ONS hostname representation as a  
 419 valid input.



420  
 421 *Figure 6 - Comparison of the data elements present in each level of each scheme*

422 **2.2. Many Schemes, 4 Levels within each scheme and multiple**  
 423 **options for each level**

424 We refer to each coding system (SGTIN, SSCC, SGLN, GRAI, GIAI and GID) as a  
 425 scheme. For the purposes of Tag Data Translation for use with Generation 1 tags, EPC  
 426 Tag Data Standards v1.1 defines separately a 64-bit scheme and a 96-bit scheme for each  
 427 of these, with the exception of the original GID, for which only a 96-bit scheme is  
 428 defined. Within each scheme, there are the four standard levels of representation as  
 429 previously described – and for some schemes, a fifth level may also be defined where the  
 430 structure of the ONS hostname is defined for that scheme. At the time of writing, this has  
 431 only been defined for SGTIN in the ONS specification.

432 Furthermore, the GS1 legacy coding schemes use a GS1 (formerly EAN.UCC) Company  
 433 Prefix of variable length, between 6 and 12 decimal digits. The TDS specification takes  
 434 two different approaches to handling this in the 64-bit and 96-bit schemes. For the 64-bit  
 435 schemes, an integer-based Company Prefix Index is encoded into the binary  
 436 representation, in order to accommodate a larger range of numbers for the Item Reference

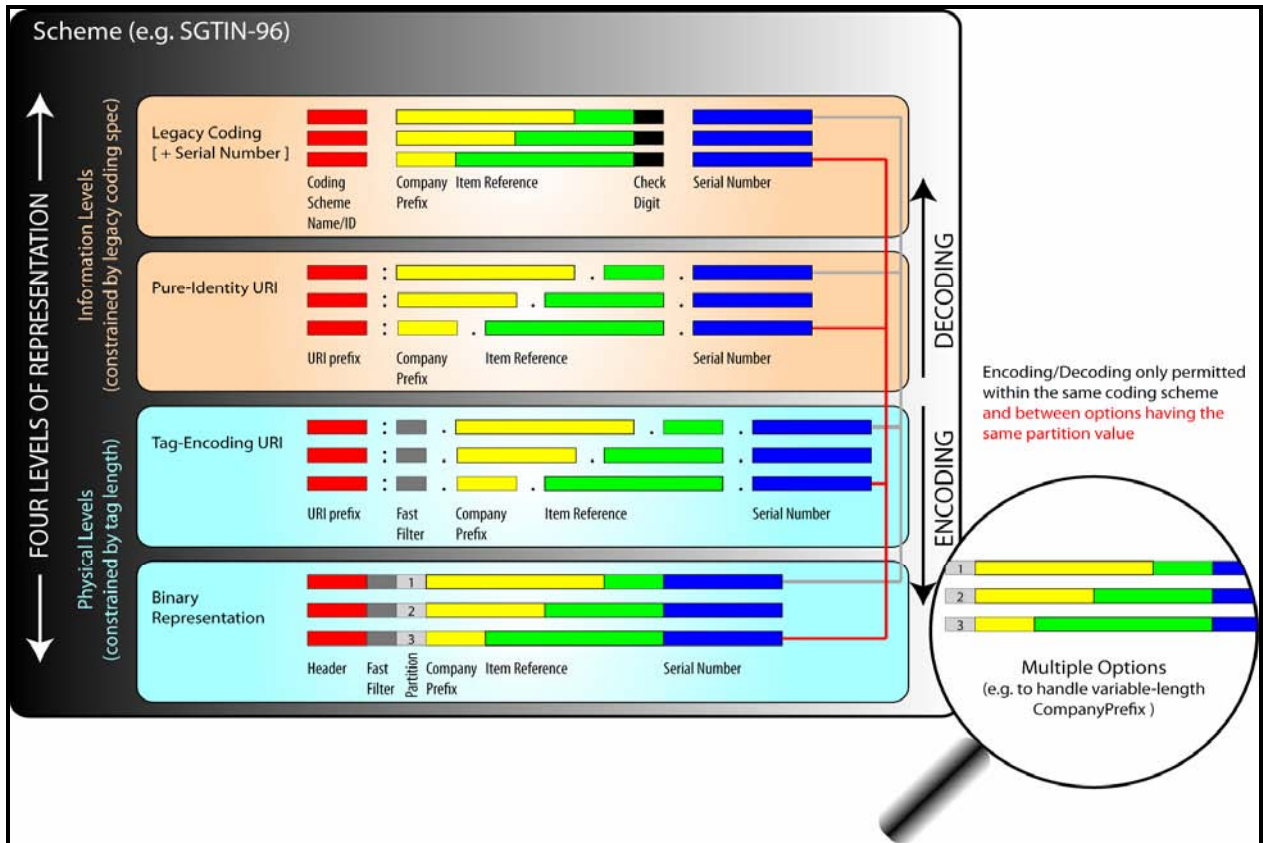
437 and Serial Number partitions. The Company Prefix is obtained from the encoded  
438 Company Prefix Index by lookup in a table and it is always the Company Prefix that  
439 appears in the URI formats. For the 96-bit schemes, a 3-bit field (the partition value)  
440 following the fast filter value within the binary representation is used to indicate the  
441 length of the Company Prefix, in the range 6-12 digits, denoted by binary partition values  
442 000 – 110. The bit-length partitions allocated to the Company Prefix and Item Reference  
443 fields varies accordingly as described in EPC Tag Data Standards.

444 One option would be to use a separate lookup table for the partition values as described in  
445 the TDS specification. However, since the correspondence between the partition value  
446 and the length of the Company Prefix is common to all the GS1 schemes and the partition  
447 table is static in nature, we propose a more pragmatic approach and instead embed 7  
448 variants ('Options') of the coding structure within each level, with the appropriate Option  
449 being selected either by matching a hard-coded partition value from the inbound data  
450 (where this is supplied in binary representation) – or from the length of the Company  
451 Prefix (which SHALL be supplied independently if encoding from the legacy coding).  
452 This approach also allows the TDT markup files to specify the length and minimum and  
453 maximum values for each field, which will often vary, depending on which Option was  
454 selected – i.e. depending on the length of the Company Prefix used.

455 For each option, the representation of the EPC is expressed as both a regular expression  
456 pattern to match the inbound representation against, and as an Augmented Backus-Naur  
457 Form (ABNF) grammar for formatting the outbound representation.

458 The regular expression patterns and ABNF grammar are therefore subtly different for  
459 each of the options within a particular level – usually in the literal values of the bits for  
460 the partition value and lengths of digits or bits for each of the subsequent partitions  
461 (where delimiters such as a period '.' separate these partitions) – or in the case of the  
462 legacy codes and binary representation, the way in which groups of digits or bits are  
463 grouped within the regular expression pattern. This approach facilitates the automatic  
464 detection of the boundary between company prefix and item reference simply by regular  
465 expression pattern matching.

466 Within each option, the various fields matched using the regular expression are specified,  
467 together with any constraints which may apply to them (e.g. maximum and minimum  
468 values), as well as information about how they should be properly formatted in both  
469 binary and non-binary (i.e. information about the number of characters or bits, when a  
470 certain length is required, as well as information about any padding conventions which  
471 are to be used (e.g. left-pad with '0' to reach the required length of a particular field). The  
472 concept of multiple options within each level of each scheme is illustrated in Figure 7.



473  
474  
475

Figure 7 - Depiction of multiple options within each level to handle variable-length Company Prefixes.

### 476 3. TDT Markup and Logical Process

477 The key element of the above architecture is the collection of TDT markup files, which  
 478 enables encoding and decoding between various levels of representation for each  
 479 particular coding scheme. This generic design requires open and highly flexible  
 480 representation of rules for translation software to encode/decode based on the input value.  
 481 The TDT markup language is a machine-readable XML format expressing the  
 482 encoding/decoding and validation rules for various identifiers / coding schemes defined  
 483 in the TDS specification. The TDT markup SHALL be created and maintained by  
 484 EPCglobal for all the identities defined by the EPC Tag Data Standard specification.

485 This chapter provides a descriptive explanation of how to interpret the TDT Markup files  
 486 in the context of a Tag Data Translation process. Chapter 4 provides a formal  
 487 explanation of the elements and attributes of the TDT markup files.

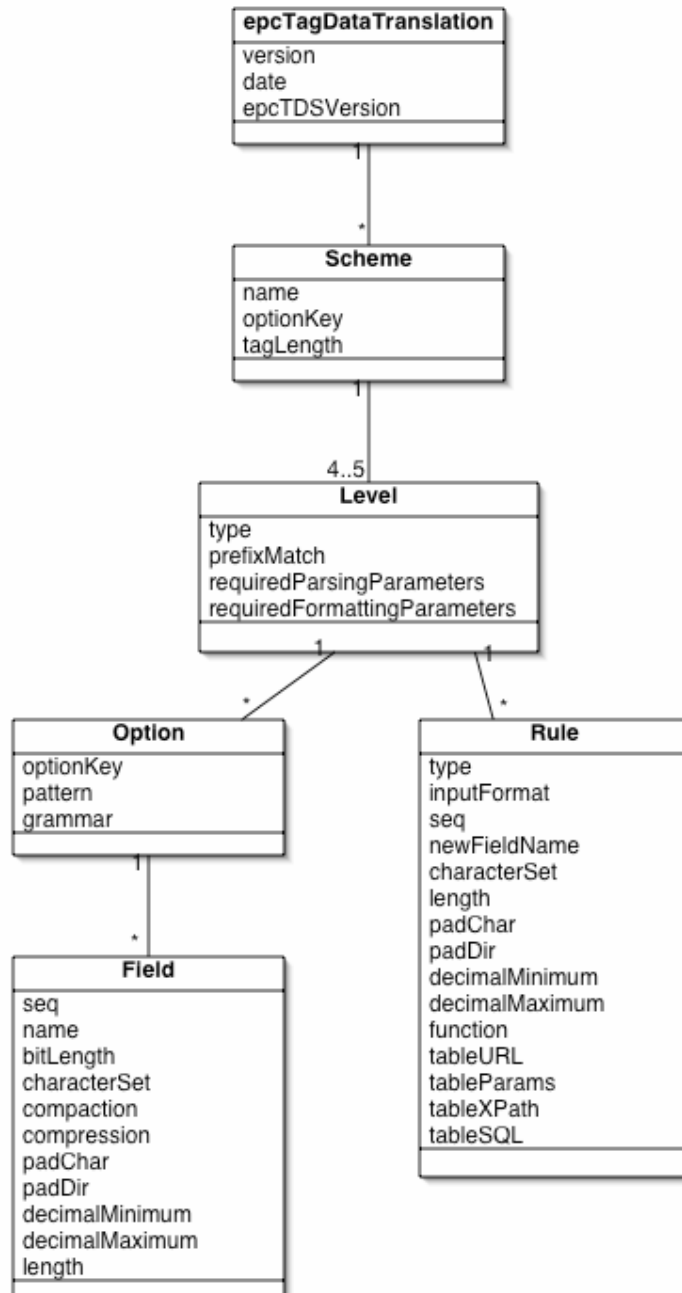
#### 488 3.1. TDT Master Index file

489 It is envisaged that separate TDT markup files or instance documents will be maintained  
 490 for each coding scheme (i.e. separate files for SGTIN-64, SGTIN-96, SSCC-64, SSCC-  
 491 96, GID-96, etc.) and referenced from a main index document also in XML, which lists  
 492 all supported coding schemes. Version control could be achieved via version numbers,

493 timestamps of updates and digests (such as MD5 digests) for each of the per-scheme  
494 instance documents referenced from the index document. Furthermore, the relevant  
495 numbering authority (such as EPCglobal) should digitally sign the relevant sections of the  
496 index document for the coding schemes over which they have authority. In this way, the  
497 integrity of the encoding/decoding instructions would be assured to have been thoroughly  
498 checked by that numbering authority and to be authoritative.

### 499 **3.2. TDT Markup**

500 The key elements of the TDT markup are shown in Figure 8.



501  
 502  
 503

Figure 8 - Tag Data Translation Markup Language schema as a UML diagram

### 504 **3.3. Definition of Formats via Regular Expression Patterns and** 505 **ABNF Grammar**

506 The TDT specification uses regular expression patterns and Augmented Backus-Naur  
507 Form (ABNF) grammar expressions to express the structure of the EPC in various levels  
508 of representation.

509 The regular expression patterns are primarily intended to be used to match the input value  
510 and extract values of particular fields via groups of bits, digits and characters which are  
511 indicated within the conventional round bracket parentheses used in regular expressions.

512 The regular expression patterns provided in the TDT markup files SHALL be written  
513 according to the Perl-Compliant Regular Expressions, with support for zero-length  
514 negative lookahead.

515 *It is not sufficient to use the XSD regexp type as documented at*  
516 *<http://www.w3.org/TR/xmlschema-2/> because it is sometimes useful to be*  
517 *able to use a negative lookahead '?' construct within the regular expressions. The*  
518 *implementations of regular expressions in Perl, Java, C#, .NET all allow for negative*  
519 *lookahead.*

520 The ABNF grammar form allows us to express the outbound string as a concatenation of  
521 fixed literal values and fields whose values are variables determined during the  
522 translation process. In the ABNF grammar, the fixed literal values are enclosed in single  
523 quotes, while the names of the variable elements are unquoted, indicating that their  
524 values should be substituted for the names at this position in the grammar. All elements  
525 of the grammar are separated by space characters. We use the Augmented Backus-Naur  
526 Form (ABNF) for the grammar rather than simple Backus-Naur Form (BNF) in order to  
527 improve readability because the latter requires the use of angle brackets around the names  
528 of variable fields, which would need to be escaped to &lt; and &gt; respectively for  
529 use in an XML document.

530 The child 'Field' elements within each option allow the constraints and formatting  
531 conventions for each individual field to be specified unambiguously, for the purposes of  
532 error-checking and validation of EPCs.

533 The use of regular expression patterns, ABNF grammar and separate nested (child) field  
534 elements with attributes for each of the fields allows for the constraints (minimum,  
535 maximum values, character set, required field length etc.) to be specified independently  
536 for each field, providing flexibility in the URI formats, so that for example an  
537 alphanumeric serial number field could co-exist alongside a decimal Company Prefix  
538 field, as would be required to support the full range of possible GRAI codes for a future  
539 tag with a larger number of bits devoted to the EPC identifier.

### 540 **3.4. Determination of the inbound representation**

541 A desirable feature of any Tag Data Translation software is the ability to automatically  
542 detect the format of the inbound string received, whether in binary, tag-encoding URI,  
543 pure-identity URI or legacy coding. Furthermore, the coding scheme should also be  
544 detected. The tag-length SHALL either be determined from the input value (i.e. given a

545 binary string or tag-encoding URI), – or otherwise, where the input value does not  
546 indicate a particular tag-length (e.g. pure-identity URI, serialized legacy code), the  
547 intended tag-length of the output SHALL be specified additionally via the supplied  
548 parameters when the input value is either a pure-identity URI or a serialized legacy code,  
549 neither of which specify the tag-length themselves. It is important that this initial  
550 matching can be done quickly without having to try matching against all possible patterns  
551 for all possible schemes, tag lengths and lengths of the Company Prefix.

552 For this reason the Tag Data Translation markup files specify a prefix-match for each  
553 level of each scheme, which SHALL match from the beginning of the input value. If the  
554 prefix-match matches, then the translation software can iterate in further detail through  
555 the full regular expression patterns for each of the options to extract parameter values –  
556 otherwise it should immediately skip to try the next possible prefix-match to test for a  
557 different scheme or different level of representation, without needing to try all the options  
558 nested within each of these, since all of the nested regular expression patterns share the  
559 same prefix-match.

560 Note that under UHF Generation 2, the bits stored in the EPC memory bank consist of 16  
561 protocol-control (PC) bits, [of which the last eight bits will normally contain either the 8-  
562 bit EPC header or an 8-bit Application Family Identifier (AFI) allocated by ISO],  
563 followed by the remainder of the EPC. The Tag Data Standards work group is currently  
564 developing a revised specification dealing with Generation 2 and upwards. The  
565 consequence for Tag Data Translation is that it may be necessary to specify  
566 independently whether input in binary or hexadecimal format was read from a Class  
567 0/Class 1 tag – or whether it was read from a Generation 2 tag. At the time of writing,  
568 the draft Reader Protocol specification is only concerned with Class 0/1 Generation 1  
569 tags.

### 570 **3.5. Specification of the outbound representation**

571 The Tag Data Translation process only permits encoding or decoding between different  
572 representations of the same scheme. i.e. it is neither possible nor meaningful to translate  
573 a GTIN into an SSCC – but within any given scheme, it is possible to translate between  
574 the four levels of representation, namely binary, tag-encoding URI, pure-identity URI and  
575 legacy coding.

576 With this constraint, it should be possible for Tag Data Translation software to perform a  
577 conversion so long as the input value and the outbound representation level are specified.

578 In addition, it would be a trivial matter for Tag Data Translation software to also provide  
579 an output format which is the hostname for which a type 35 ('NAPTR') DNS lookup  
580 should be made in order to effect an ONS query. Note that this is an output-only  
581 representation, as indicated in Table 2.

### 582 **3.6. Specifying supplied parameter values**

583 Decoding from the binary level through the tag-encoding URI, pure-identity URI and  
584 finally to the legacy coding strings only ever involves a potential loss of information.  
585 With the exception of the lookup table mapping Company Prefix Index to Company



586 Prefix for the 64-bit tags, it is not necessary to specify supplied parameters when  
 587 decoding, since the binary and tag-encoding formats already contain more information  
 588 than is required for the pure-identity or legacy coding formats.

589 Encoding often requires additional information to be supplied independently of the  
 590 inbound string. Examples of additional information include:

- 591 • Independent knowledge of the length of the Company Prefix
- 592 • Intended length of the physical tag (64-bit, 96-bit ...) to be encoded
- 593 • Fast filter values (e.g. to specify the packaging type – item/case/pallet)

594 It should be possible to provide these supplied parameters to Tag Data Translation  
 595 software. In all the cases above, this may simply populate an internal key-value lookup  
 596 table or associative array with parameter values additional to those that are automatically  
 597 extracted from parsing the inbound string using the matching groups of characters within  
 598 the appropriate matching regular expression pattern.

599 Note that for legacy codes such as GTIN and GLN, which are extended with serial  
 600 numbers for EPC use, the serial number SHALL NOT be passed via the supplied  
 601 parameters. Instead, the serial number SHALL be passed as part of the input value. In  
 602 this way, either the GTIN or GLN and the serial number CAN be obtained as the output  
 603 value because the same grammar is used for both input and output. This is important  
 604 because the Tag Data Translation Application Programming Interface (API) defined in  
 605 Chapter 6 of this document provides no direct access to the private values of intermediate  
 606 variables or fields used within the translation process. Table 3 shows examples of how  
 607 the input value should be formatted for serialized legacy codes. Note that SSCC, GRAI  
 608 and GIAI are already intrinsically serialized and should not be appended with  
 609 ';serial=...'.  
 610

Coding Scheme	Example string format for input of legacy codes
SGTIN	gtin=00037000302414;serial=10419703
SSCC	sscc=000370003024147856
SGLN	gln=0003700030241;serial=1041970
GRAI	grai=00037000302414274877906943
GIAI	giai=00370003024149267890123
GID	generalmanager=5;objectclass=17;serial=23
USDOD	cageordodaac=AB123;serial=3789156

611  
 612 *Table 3 Example formats for supplying serialized legacy codes as the input value.*  
 613

614 Note that in Tag Data Translation implementations, the values extracted from the  
 615 inbound EPC representation SHALL always override the values extracted from the

616 supplied parameters; i.e. the parameter string may specify 'filter=5' – but if the  
617 inbound EPC representation encodes a fast filter value of 3, then the value of 3 shall be  
618 used for the output since the value extracted from the input value overrides any values  
619 supplied via the supplied parameters.

620 Although many programming languages support the concept of an associative array as a  
621 data type, these are not generally portable across different languages in the way that data  
622 types such as integer and string are. For this reason, the associative array of key-value  
623 pairs for the supplied parameters SHALL be passed as a string format, using a semicolon  
624 [;] as the delimiter between multiple key=value pairs. A string in this format can be  
625 readily converted into an associative array in most modern programming languages,  
626 while remaining portable and language-unspecific.

### 627 **3.7. Validation of values for fields and fields derived via rules**

628 The `field` element and the `rule` element contain several attributes for validating and  
629 ensuring that the values for particular fields fall within valid ranges, both in terms of  
630 numeric ranges, as well as lengths of characters, allowed character ranges and the use of  
631 padding characters.

632 TDT markup files use such an explicit markup of the format and constraints of each field  
633 in order to provide for a great deal of future extensibility, particularly for encoding  
634 alphanumeric characters, already required by the US DOD formats.

### 635 **3.8. Restricting and checking decimal ranges for values of fields**

636 In some cases, the numeric range which can be expressed using the specified number of  
637 bits exceeds the maximum decimal value permitted by the corresponding legacy coding  
638 specifications.

639 For example, the serial number of an SSCC may be up to ten decimal digits – permitting  
640 the decimal numbers 1-9,999,999,999. This requires 34 bits to encode in binary.  
641 However, 34 bits would allow numbers in the range 0-17,179,869,183, although those  
642 between 10,000,000,000 and 17,179,869,183 are deemed not valid for use as the serial  
643 reference of an SSCC – and should result in an error if an attempt is made to encode these  
644 into an SSCC.

645 In order to prevent encoding of numbers outside the ranges permitted by the legacy  
646 coding specifications, the decimal minimum and decimal maximum limits of each field  
647 are indicated via the field attributes `decimalMinimum` and `decimalMaximum`.  
648 Where these attributes are omitted, no numeric (minimum,maximum) limits are specified  
649 and checking of numeric range NEED NOT be performed by TDT implementations.  
650 Otherwise, where numeric values are specified, the software should check that the value  
651 of the field lies within the inclusive range, i.e.

652 
$$\text{decimalMinimum} \leq \text{field} \leq \text{decimalMaximum}$$

653 Values which fall outside of the specified range should throw an exception.

### 654 **3.9. Restricting and checking character ranges for values of** 655 **fields**

656 The `characterSet` attribute of the `field` element indicates the allowed range of  
657 characters which may be present in that field. The range is expressed using the same  
658 square-bracket notation as for character ranges within regular expressions. The asterisk  
659 symbol following the closing square bracket indicates that 0 or more characters within  
660 this range are required to match the field in its entirety. Implementations may find it  
661 useful to add a leading caret (^) and a trailing dollar symbol (\$) to ensure that the  
662 `characterSet` matches the entire field. e.g. for `[0-7]*` in the TDT markup, TDT  
663 implementations may use `^[0-7]*$` as the regular expression pattern.

664 *For example,*

665 `[01]*` *permits only characters '0' and '1'*

666 `[0-7]*` *permits only characters '0' thru '7' inclusive*

667 `[0-9]*` *permits only characters '0' thru '9' inclusive*

668 `[0-9 A-Z\-*]` *permits digits '0' thru '9', the SPACE character (ASCII 32) and upper-case*  
669 *letters 'A' thru 'Z' inclusive and the hyphen character.*

670

671 The `characterSet` attribute allows checking that all of the characters fall within the  
672 permitted range. For example, if a user specifies a serial number for GRAI containing  
673 characters that are not wholly numeric, although the character ranges for GRAI-96 and  
674 GRAI-64 only permit wholly numeric serial numbers, i.e. characters in the range [0-9],  
675 this should result in an error. Note however that an error might not be reported in the  
676 situation where a user attempts to encode an alphanumeric GRAI serial code onto a 96-bit  
677 tag in the case where the serial code supplied fortuitously happens not to contain any  
678 alphabetic characters.

679 Furthermore, a 256-bit GRAI EPC might exist with two headers – one for wholly  
680 numeric serial numbers, the other for alphabetic serial numbers. The presence of the  
681 `compaction` attribute SHALL indicate that a particular field is to be interpreted as the  
682 binary encoding of a character string; its absence SHALL indicate that the field should be  
683 interpreted as an integer value or all-numeric integer string, with leading pad characters if  
684 the `padChar` attribute is also present and the integer has fewer digits than the `length`  
685 attribute specifies.

686 Tag Data Translation software SHOULD NOT rely upon particular values of the  
687 `characterSet` attribute as an alternative to taking notice of the `compaction`  
688 attribute; certain coding schemes, such as the US DOD's CAGE code omit certain  
689 characters, such as the letter 'I' in order to reduce confusion with the digit '1', when the  
690 CAGE code is communicated in human-readable format – in this case, the  
691 `characterSet` attribute may look like `'[0-9 A-HJ-NP-Z]*'`, in which case a naïve  
692 search for 'A-Z' in the `characterSet` attribute would fail to match, even though the  
693 binary value SHOULD BE converted to a character string because the `compaction`  
694 attribute was present.

### 695 **3.10. Padding of fields**

696 Certain fields within either the binary representation, the URI representations and also the  
697 legacy codes require the padding of the value to a particular number of characters, digits  
698 or bits, in order to reach a particular length for that field.

699 The attributes `length`, `padDir` and `padChar` MAY appear within any `field` or  
700 `rule` element of the TDT markup files. Within each `field` element, all three SHALL  
701 either be present together – or all three SHALL be absent together. Within `rule`  
702 elements, there is no requirement for the `padDir` and `padChar` attributes to be present,  
703 even if the `length` attribute is specified; functions defined in rules may return a value  
704 which does not require further padding – in this case, the `length` attribute may be  
705 specified, merely in order to verify that the result is of the correct length of characters.

706 If the `padChar` attribute is absent, then the field value or new field value obtained via  
707 execution of the rule SHALL NOT be padded.

708 If the `padChar` attribute is present, then the non-binary field value or new non-binary  
709 field value obtained via execution of the rule SHALL be padded according to the values  
710 of the attributes `length`, `padDir` and `padChar`. At the binary level of representation,  
711 each field SHALL first be converted to binary, then padded with zero bits to reach the  
712 number of bits indicated by the `bitLength` attribute for that field.

713 The `padChar` attribute SHALL consist of a single character to be used for padding.  
714 Typically this is the '0' digit (ASCII character 48 [30 hex]). Other coding schemes MAY  
715 specify the space character (ASCII character 32 [20 hex]) or a different character to use.  
716 The character indicated by the `padChar` attribute SHALL only be used for padding the  
717 non-binary equivalents of fields. When fields are to be encoded into the binary format,  
718 they SHALL always be padded with zero ['0'] bits as required, in order to reach the  
719 number of bits indicated by the `bitLength` attribute for that field.

720 The `padDir` attribute SHALL take a string value of either 'LEFT' or 'RIGHT', indicating  
721 whether the pad characters should appear to the left or right of the unpadded value. The  
722 `padDir` attribute is 'LEFT' for all coding schemes specified in Tag Data Standards v1.1.

723 For all levels except the binary level, if a `field` or `rule` element contains a `padChar`  
724 attribute, then the field SHALL be padded with repetitions of the character indicated by  
725 the `padChar` attribute, in the direction indicated by `padDir` attribute so that the padded  
726 value of the field has the length of characters as specified by the `length` attribute. This  
727 applies at the validation, parsing, rule execution and formatting stages of the translation  
728 process.

729 For the binary level, each field element SHALL contain a `bitLength` attribute and  
730 MAY contain a `length` attribute.

731 At the binary level, when encoding (formatting) into binary, the field value SHALL first  
732 be converted to binary, then padded to the appropriate number of zero ['0'] bits indicated  
733 by the `bitLength` attribute.

734 If the `padChar` attribute is present, then upon decoding (parsing or extracting) a field  
735 from the input value, its non-binary equivalent SHALL afterwards be padded with

736 repetitions of the character indicated by the `padChar` attribute, in the direction indicated  
737 by `padDir` attribute so that the padded non-binary value of the field has the length of  
738 characters as specified by the `length` attribute.

739 *An example is the GSI Company Prefix field in a 96-bit EPC – upon conversion from*  
740 *binary to decimal, the length attribute indicates the required length, so that the original*  
741 *Company Prefix can be recovered as a numeric string with leading zeros, e.g. '0037000'.*

742 If the `padChar` attribute is absent, then upon decoding a field from the input value, its  
743 non-binary equivalent SHALL NOT be padded.

744 *An example is the Company Prefix Index field in a 64-bit EPC – upon conversion from*  
745 *binary to decimal, the absence of the length attribute indicates that the binary value*  
746 *should not be padded. Instead, it is converted to an unpadded integer value, which is*  
747 *used as the key for a TABLELOOKUP function specified in a rule of type 'EXTRACT', in*  
748 *order to look up the corresponding GSI Company Prefix.*

749 These rules about whether or not to pad are summarized in Table 4

		padChar ABSENT	padChar PRESENT
BINARY LEVEL	Parsing/Extracting From Binary (DECODING)	<p>If <code>compaction</code> attribute is ABSENT, Convert field from binary to integer and DO NOT PAD. Integer should have no leading zeros.</p> <p>If <code>compaction</code> attribute is PRESENT, convert field from binary to character string using appropriate compaction scheme method and DO NOT PAD this further.</p>	<p>If <code>compaction</code> attribute is ABSENT, Convert field from binary to integer THEN PAD with <code>padChar</code> characters in <code>padDir</code> direction to reach a numeric string of <code>length</code> characters.</p> <p>If <code>compaction</code> attribute is PRESENT, convert field from binary to character string using appropriate compaction scheme method THEN PAD with <code>padChar</code> characters in <code>padDir</code> direction to reach a character string of <code>length</code> characters..</p>
	Formatting Output To Binary (ENCODING)	<p>CONVERT field to binary string (using appropriate compaction scheme method if <code>compaction</code> attribute is specified) THEN PAD binary field to the left with 'zero' bits to reach <code>bitLength</code> bits</p>	<p>PAD field with <code>padChar</code> characters in <code>padDir</code> direction to reach a character string of <code>length</code> characters THEN CONVERT field to binary string (using appropriate compaction scheme method if <code>compaction</code> attribute is specified) THEN PAD binary field to the left with 'zero' bits to reach <code>bitLength</code> bits</p>
ALL OTHER LEVELS	Parsing/Extracting AND Formatting Output	DO NOT PAD	PAD with <code>padChar</code> characters in <code>padDir</code> direction to reach a string of <code>length</code> characters.

*Table 4 – Summary of rules about whether or not to pad a field*

753 *For example, for a 96-bit SGTIN, for the field whose name="companyprefix", the*  
754 *length attribute is 7, the padChar attribute is '0' and the padDir attribute is 'LEFT'*  
755 *for the option where optionKey = 7. This means that a 24-bit binary value of*  
756 *'000000001001000010001000' read from the tag for the field named companyprefix*  
757 *should be converted to the integer 37000, then padded with the pad character '0' to 7*  
758 *characters, yielding '0037000' as the numeric string value for this field.*

759

760 *For a SGLN where the length of the companyprefix is 12 digits, the location reference is*  
761 *a string of zero characters length. This may result in URIs which look strange because*  
762 *there is an empty string between two successive delimiters, e.g. '..' in a URL which looks*  
763 *like urn:epc:id:sgln:123456789012..12345*

764 *This is however correct – and it is incorrect to render the zero-length field as '0' between*  
765 *the period (.) delimiters because '0' is of length 1 character – not zero characters length*  
766 *as required by the length attribute of the appropriate <field> element.*

### 767 **3.11. Compaction and Compression of fields**

768 When strings other than purely numeric strings are to be encoded in the binary level of  
769 representation, the field element contains two additional attributes, compaction and  
770 compression. Absence of the compaction attribute SHALL indicate that the  
771 binary value represents an integer or all-numeric string. Presence of the compaction  
772 attribute SHALL indicate that the binary value represents a character string encoded into  
773 binary using a per-character compaction method for economizing on the number of bits  
774 required. Allowed values are '5-bit', '6-bit', '7-bit' and '8-bit', referring to the  
775 compaction methods described in ISO 15962, in which the most significant 3/2/1/0 bits of  
776 the 8-bit ASCII byte for each character are truncated.

777 Note that a compaction value of '8-bit' SHALL be used to indicate that each  
778 successive eight bits should be interpreted as an 8-bit ASCII character, even though there  
779 is effectively no compaction or per-byte truncation involved, unlike the other values of  
780 the compaction attribute. The compaction values '16-bit' and '32-bit' are not used  
781 in the markup files for this version of the TDT specification – but are reserved in the  
782 TDT XSD schema and SHALL indicate 16-bit and 32-bit UNICODE representation  
783 where this is required in the future.

784 The compression attribute is intended for future use, to indicate a compression  
785 technique to be applied to the value as a whole, rather than on a per-character basis.  
786 Permitted values for the compression attribute are not currently defined in this  
787 version of the Tag Data Translation specification but those values defined in future may  
788 indicate compression techniques such as zip / gzip compression, Huffman encoding etc.

### 789 **3.12. Names of fields used within the TDSv1.1r1.27 schemes**

790 The names of fields appearing in the TDT markup files are completely arbitrary but by  
791 convention SHALL consist of lower case alphanumeric words with no spaces or hyphens.  
792 There are no reserved words and the use of a name within one coding scheme does not

793 imply any correlation with an identically named field within a different coding scheme;  
 794 each coding scheme effectively has its own namespace for field names. Table 5 lists  
 795 some field names that are used in the coding schemes for EPC Tag Data Standards  
 796 v1.1r1.27

filter	fast filter value – decimal range 0-7
serial	serial number – decimal or alphanumeric
companyprefixlength	length of a GS1 company prefix as a number of characters – decimal e.g. company prefix '0037000' → companyprefixlength=7
taglength	64/96/256 etc. – number of bits for the EPC identifier
companyprefixindex	an integer-based lookup key for accessing the real Company Prefix – for use with 64-bit tags
itemref	Identifies the Object Type or SKU within a particular company for a GTIN
locationref	Identifies the Location within a company for a GLN
assetref	A serialised asset reference – for use with the GIAI
serialref	A serialised reference – e.g. for use with the SSCC

797 *Table 5 – Names of fields used within Tag Data Standards v1.1 r1.27*

### 798 **3.13. Rules and Derived Fields**

799 Certain fields required for formatting the outbound representation are not obtained simply  
 800 from pattern matching of the inbound representation. A sequence of rules allows the  
 801 additional fields to be derived from fields whose values are already known.

802 The reason why this is necessary is that there is often some manipulation of the legacy  
 803 codes required in order to translate them into the pure-identity URI representation.  
 804 Examples include string manipulation such as the relocation of the initial indicator digit  
 805 or extension digit to the front of the item reference field – or for decoding, the re-  
 806 calculation of the GS1 checksum – and appending this as the last digit of the legacy  
 807 coding representation. Likewise, replacement of the Company Prefix Index integer by  
 808 the corresponding GS1 Company Prefix is something that is not readily expressed simply  
 809 via regular expressions. By working through an example for the GTIN, it is clear that  
 810 although the processing steps are reversible between encoding into the pure-identity URI  
 811 and decoding into the legacy codes, the way in which those steps are defined takes on an  
 812 unsymmetrical appearance in the sequence of rules. An example illustrates this point:  
 813



814 Decoding the GTIN (i.e. translating from pure-identity URI into legacy  
815 coding)

816

- 817 • `indicatordigit = SUBSTR(itemref,0,1);`
- 818 • `itemrefremainder = SUBSTR(itemref,1);`
- 819 • `gtinprefix =`  
820 `CONCAT(indicatordigit,companyprefix,itemrefremainder);`
- 821 • `checkdigit = GS1CHECKSUM(gtinprefix);`

822

823 The above are all examples of rules to be executed at the 'EXTRACT' stage, i.e.  
824 immediately after parsing the input value.

825

826 Encoding the GTIN (i.e. translating from legacy coding into pure-  
827 identity URI)

828 (assumes `companyprefixlength` is passed as a supplied parameter)

829

- 830 • `gtinprefixremainder=SUBSTR(gtin,1,12);`
- 831 • `indicatordigit=SUBSTR(gtin,0,1);`
- 832 • `itemrefremainder=SUBSTR(gtinprefixremainder,companyprefix`  
833 `length);`
- 834 • `itemref=CONCAT(indicatordigit,itemrefremainder);`
- 835 • `companyprefix=SUBSTR(gtinprefixremainder,0,companyprefixl`  
836 `ength);`

837

838 The above are all examples of rules to be executed at the 'FORMAT' stage, i.e. when  
839 constructing the output value.

840

841 As the above examples show, the definitions of particular fields (e.g. `itemrefremainder`)  
842 depends upon whether encoding or decoding is being performed (or equivalently,  
843 whether the field is required for formatting the output value – or being extracted from the  
844 input value), since each successive definition depends on prior execution of the  
845 definitions preceding it, in the correct order, in order that all the required fields are  
846 available.

847 The rules in the example above apply generally, with minor modifications to all of the  
848 GS1 coding schemes covered in the TDS Specification v1.1r1.27. It is worth noting that

849 each of the above rule steps contains only one function or operation per step, which  
 850 means that even a very simple parser can be used, without needing to deal with nesting of  
 851 functions in parentheses.

### 852 3.14. Core Functions

853 The core functions which SHALL be supported by Tag Data Translation software in  
 854 order to encode/decode the GS1 coding schemes are described in Table 6.

SUBSTR (string, offset)	the substring starting at <offset> (offset=0 is the first character of string)
SUBSTR (string, offset, length)	the substring starting at <offset> (offset=0 is the first character of string) and of <length> characters
CONCAT (string1, string2, string3,...)	concatenation of string parameters
LENGTH(string)	number of characters of a string
GS1CHECKSUM (string)	Computes the GS1 checksum digit given a string containing all the preceding digits
TABLELOOKUP (inval, tablename, incol, outcol)	Performs a lookup in table called tablename. Given an input value <inval>, look in table <tablename> to find a match in column names <incol> and return the corresponding value for the same row from output column <outcol>. The TABLELOOKUP function only indicates the logical lookup – not any bindings. The table URL is specified via a separate attribute tableURL and bindings to XPath or SQL expressions are specified via separate attributes tableXPath and tableSQL.

855

856 *Table 6 - Basic built-in functions required to support encoding and decoding within the*  
 857 *GS1 schemes currently covered by the TDS specification*

858

859 In order to make full use of the Tag Data Translation markup files, implementations of  
 860 translation software should provide equivalent functions in the programming language in  
 861 which they are written, either by the use of native functions or custom-built methods,  
 862 functions or subroutines.

863 In this version of Tag Data Translation, the requirement that implementations should be  
 864 able to recalculate check digits only applies to the GS1 coding schemes, when output in  
 865 the legacy format is required. Further details on calculation of the GS1 checksum can be

866 found at <http://www.gs1.org>. It should be noted that ISO 7064 provides a  
 867 standard for more general-purpose calculation of check digits and that this may be  
 868 considered in future versions of this specification.

869 It is important to note that modern programming languages (including Java, C++, C#,  
 870 Visual Basic, Perl, Python) do not all share the same convention in the definitions of their  
 871 native functions, especially for string functions. In some languages the first character of  
 872 the string has an index 0, whereas in others, the first character has an index 1.  
 873 Furthermore, many of the languages provide a substring function which takes two  
 874 additional parameters as well as the string itself. Usually, the first of these is the start  
 875 index, indicating the starting position where the substring should be extracted. However,  
 876 some languages (e.g. Java, Python) define the last parameter as the end index, whereas  
 877 others (C++, VB.Net, Perl) define it as the length of the substring, i.e. number of  
 878 characters to be extracted. Table 7 indicates a number of language-specific equivalents  
 879 for the three-parameter SUBSTR function in Table 6.

880

	SUBSTR(string, offset, length)	Notes
C++	String.substr(offset, length);	
C#	String.Substring(offset, length);	
Perl	substr(\$stringvariable, offset, length);	
Visual Basic	String.Substring(offset, length)	
Java	Java.lang.String String.substring(beginIndex, endIndex)	beginIndex = offset endIndex = offset+length
Python	String[start:end]	start = offset end = offset+length

881 *Table 7 – Comparison of how substring functions are defined in a number of modern*  
 882 *programming languages. The parameters offset and length are of integer type.*

883

884 Note that for the case of rules which use the TABLELOOKUP function, additional  
 885 attributes tableURL and tableXPath or tableSQL are provided. Tables may be  
 886 provided in XML format or as comma-separated values (CSV) or tab-separated values  
 887 (TSV), even though any Tag Data Translation software MAY internally store the table  
 888 values in a different format altogether. For this reason, the binding to the original format  
 889 is handled separately via the tableURL and tableParams and either  
 890 tableXPath or tableSQL attributes, while the TABLELOOKUP function expresses  
 891 the logical lookup, irrespective of the format in which any table is actually supplied.

892

893 As an example, consider the GS1 Company Prefix Index lookup tables for use with 64-bit  
894 tags. An XML version and a comma-separated values (CSV) version are provided at  
895 <http://www.onsepc.com>

896

897 For the XML version,  
898 `tableURL="http://www.onsepc.com/ManagerTranslation.xml"` and  
899 `tableXPath` and `tableParams` are one of the following pairs:

900

901 `tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"`

902 `tableParams="companyprefixindex"`

903 for the case where

904 `function="TABLELOOKUP(companyprefixindex, 'GEPC64Table', comp`  
905 `anyprefixindex, companyprefix) "`

906 OR

907 `tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index"`

908 `tableParams="companyprefix"`

909 for the case where

910 `function="TABLELOOKUP(companyprefix, 'GEPC64Table', companypr`  
911 `efix, companyprefixindex) "`

912

913 The first example pair is used to obtain the value of `companyprefix` given the value  
914 of `index` (e.g. retrieve `companyprefix='0037000'` given  
915 `companyprefixindex='1'`).

916 The second example pair is used to obtain the value of `companyprefixindex` given  
917 the `company prefix` (e.g. retrieve `companyprefixindex='1'` given that  
918 `companyprefix='0037000'`).

919 Note that `tableParams` may be a comma-separated string of either fieldnames (if  
920 unquoted) or fixed literal values, if wholly numeric or single-quoted strings. The `$1` in  
921 the `tableXPath` expressions indicates that the actual value of the field named by the  
922 first parameter of `tableParams` string should be substituted into the `tableXPath`  
923 expression at this point before passing the XPath expression to an XML DOM parser.

924 For example, if the value of `companyprefix` is '0037000', then for the second example  
925 pair, the value of '0037000' would be substituted in place of '\$1' in `tableXPath` so that  
926 it would be the following XPath expression:

927 `"/GEPC64Table/entry[@companyPrefix='0037000']/@index"`

928 which is actually passed to the XML DOM parser.

929

930 Where more than one parameter is listed in `tableParams`, `$2` indicates where to  
931 substitute the second parameter, while `$3` indicates where to substitute the third  
932 parameter, and so on.

933

934 A table supplied as comma-separated values (CSV) or tab-separated values (TSV), can be  
935 readily converted to a relational database table with the appropriate column headings.

936 For the example of the Company Prefix Index table for 64-bit tags, the CSV version is  
937 available from `http://www.onsepc.com/ManagerTranslation.csv`

938 In this case, the attribute  
939  
940 `tableURI= "http://www.onsepc.com/ManagerTranslation.csv"`  
941  
942 and the attributes `tableSQL` and `tableParams` may be one of the following pairs:  
943

944 `tableSQL="SELECT companyPrefix from GEPC64Table WHERE`  
945 `index= '$1' "`

946 `tableParams="companyprefixindex"`

947 for the case where

948 `function="TABLELOOKUP(companyprefixindex, 'GEPC64Table', comp`  
949 `anyprefixindex, companyprefix) "`

950 OR

951 `tableSQL="SELECT index from GEPC64Table WHERE`  
952 `companyPrefix= '$1' "`

953 `tableParams="companyprefix"`

954 for the case where

955 `function="TABLELOOKUP(companyprefix, 'GEPC64Table', companypr`  
956 `efix, companyprefixindex) "`

957

958 Each of the two example pairs above corresponds to the respective pairs in the previous  
959 examples for the `tableXPath` attributes. Likewise, the notation `$1`, `$2`, etc.  
960 indicates where values of fields named by parameters from the `tableParams` string  
961 should be substituted into the `tableSQL` expression before passing to the relational  
962 database engine for execution.

## 963 4. TDT Markup - Elements and Attributes

### 964 4.1. Root Element

965 The epcTagDataTranslation element is the root element of the TDT definition.

#### 966 4.1.1. Attributes

Name	Description	Example Values
version	TDT Definition version number	1.00
date	Creation Date	2005-03-07T11:33Z
epcTDSVersion	TDS Specification version	1.1r1.27

#### 967 4.1.2. Elements

Name	Description
scheme	Please see scheme definition below for more details

### 968 4.2. Scheme Element

969 For every identifier / coding scheme as defined in the TDS specification, the Scheme  
970 element provides details of encoding/decoding rules and formats for use by Tag Data  
971 Translation software. In this version of the TDT specification, markup files are provided  
972 for the following identifiers: SGTIN-64, SGTIN-96, SSCC-64, SSCC-96, GRAI-64,  
973 GRAI-96, GIAI-64, GIAI-96, SGLN-64, SGLN-96 and GID-96.

#### 974 4.2.1. Attributes

Name	Description	Example Values
name	Name of the coding scheme	SGTIN-64, SGTIN-96, SSCC-64, SSCC-96, GRAI-64, GRAI-96, GIAI-64, GIAI-96, SGLN-64, SGLN-96 and GID-96
optionKey	The name of a variable whose value determines which one of multiple options to select	companyprefixlength
tagLength	Tag length	64 or 96

975 **4.2.2. Elements**

Name	Description
Level	Contains option elements expressing a pattern, grammar and encoding/decoding rules for each level of representation

976 **4.3. Level Element**

977 This element provides a prefix match for each level of representation. Nested within the  
 978 level element are option elements (which provide the pattern regular expressions  
 979 for parsing the input into fields and ABNF grammar for formatting the output) and  
 980 rule elements used for obtaining additional fields from functional operations on known  
 981 fields.

982 **4.3.1. Attributes**

Name	Description	Example Values
type	Indicates level of representation	BINARY TAG_ENCODING PURE_IDENTITY LEGACY ONS_HOSTNAME
prefixMatch	Prefix value required for each encoding/decoding level	00001010 uri:epc:tag:sscc-64 uri:epc:id:sscc sscc=
requiredParsingParameters	Comma-delimited string listing names of fields whose values SHALL be specified in the list of suppliedParameters in order to parse the fields of an input value at this level	companyprefixlength
requiredFormattingParameters	Comma-delimited string listing names of fields whose values SHALL be specified in the list of suppliedParameters in order to format the outbound value at this level	filter,taglength

983 **4.3.2. Elements**

Name	Description
option	Contains patterns and grammar
rule	Contains rules required for determining values of additional variables required

984

985 **4.4. Option Element**

986 **4.4.1. Attributes**

Name	Description	Example Values
optionKey	A fixed value which the optionKey attribute of the Scheme element SHALL match if this option is to be considered	any string value but for GS1 legacy codes defined in TDS v1.1, the values '6','7','8','9','10','11','12'
pattern	A regular expression pattern to be used for parsing the input string and extracting the values for variable fields	00101111([01]{4})00100000([01]{40})([01]{36})
grammar	An ABNF grammar indicating how the output can be reassembled from a combination of literal values and substituted variables (fields)	'00101111' filter cageordodaac serial  <i>N.B. single quoted string indicate fixed literal strings, unquoted strings indicate substitution of the correspondingly named field values</i>

987 **4.4.2.**



988 **4.4.3. Elements**

Name	Description
field	Provides information about each of the variables, e.g. (min, max) values, allowed character set, length, padding etc.

989 **4.5. Field Element**

990 **4.6. Attributes**

Name	Description	Example Values
seq	The sequence number for a particular sub-pattern matched from a regular expression – e.g. 1 denotes the first sub-pattern extracted	1, 2, 3...
name	The name of the variable (or field) – just a reference used to ensure that each field may be used to construct the output format	filter, companyprefix, itemref, serial, ...
decimalMinimum	Decimal minimum value allowed for this field	0
decimalMaximum	Decimal maximum value allowed for this field	9999999
length	Required length of this field in string characters.	7
bitLength	Required length of this field in bits. Omitted for the non-binary levels.	24
characterSet	Allowed character set for this field, expressed in regular expression character range notation	[0-9],[01]
padChar	Character to be used to pad to required value of fieldlength. Omitted if no padding is required for the non-binary form. Presence indicates padding is required for the non-binary form.	'0', ''
padDir	Direction to insert pad characters.	'LEFT', 'RIGHT'

991 **4.7. Rule Element**

992 **4.7.1. Attributes**

Name	Description	Example Values
type	Indicates at which stage of the process the definition should be evaluated	EXTRACT, FORMAT
inputFormat	Indicates whether the input parameter to the definition is in binary format or non-binary ('string') format	STRING, BINARY
seq	A sequence number to indicate the running order for definitions sharing the same mode value. The definitions should be run in order of ascending 'seq' value	1,2,3,4,5...
newFieldName	A name for the new field or variable whose value is determined by evaluating the definition	Any string consisting of alphanumeric characters and underscore
function	An expression indicating how the new field can be determined from a function of already-known fields	e.g. SUBSTR(itemref,0,1)
decimalMinimum	For numeric fields, the decimal minimum value allowed for this field	e.g. 0
decimalMaximum	For numeric fields, the decimal maximum value allowed for this field	e.g. 9999999
length	Required length of this field in string characters.	7
padChar	Character to be used to pad to required value of fieldlength. Omitted if no padding is required. Present if padding is required.	'0', ''
padDir	Direction to insert pad characters	'LEFT', 'RIGHT'

Name	Description	Example Values
characterSet	Allowed character set for this field, expressed in regular expression character range notation	[0-9],[01]
tableURL	A URL where the data table can be obtained	http://www.onsepc.com/ManagerTranslation.xml
tableXPath	<p>An XPath expression for obtaining a particular attribute or element value from an XML table.</p> <p>The inline notation '\$1', '\$2' etc. indicates where the values of the first, second, etc. elements of the tableParams list should be substituted before passing to an XML parsing engine.</p>	/GEPC64Table/entry[@index='\$1']/@companyPrefix
tableSQL	<p>A SQL expression for obtaining a particular field from a relational database table.</p> <p>The inline notation '\$1', '\$2' etc. indicates where the values of the first, second, etc. elements of the tableParams list should be substituted before passing to a relational database query engine.</p>	SELECT companyPrefix FROM GEPC64Table WHERE index='\$1'
tableParams	A comma-delimited string list of fieldnames whose actual values should be substituted into the tableXPath or tableSQL expressions	e.g. companyprefixindex

993 **5. Translation Process**

994

995 The execution of the rules in the TDT process takes place at two distinct processing  
996 stages, denoted 'FORMAT' and 'EXTRACT', as explained in Table 8:

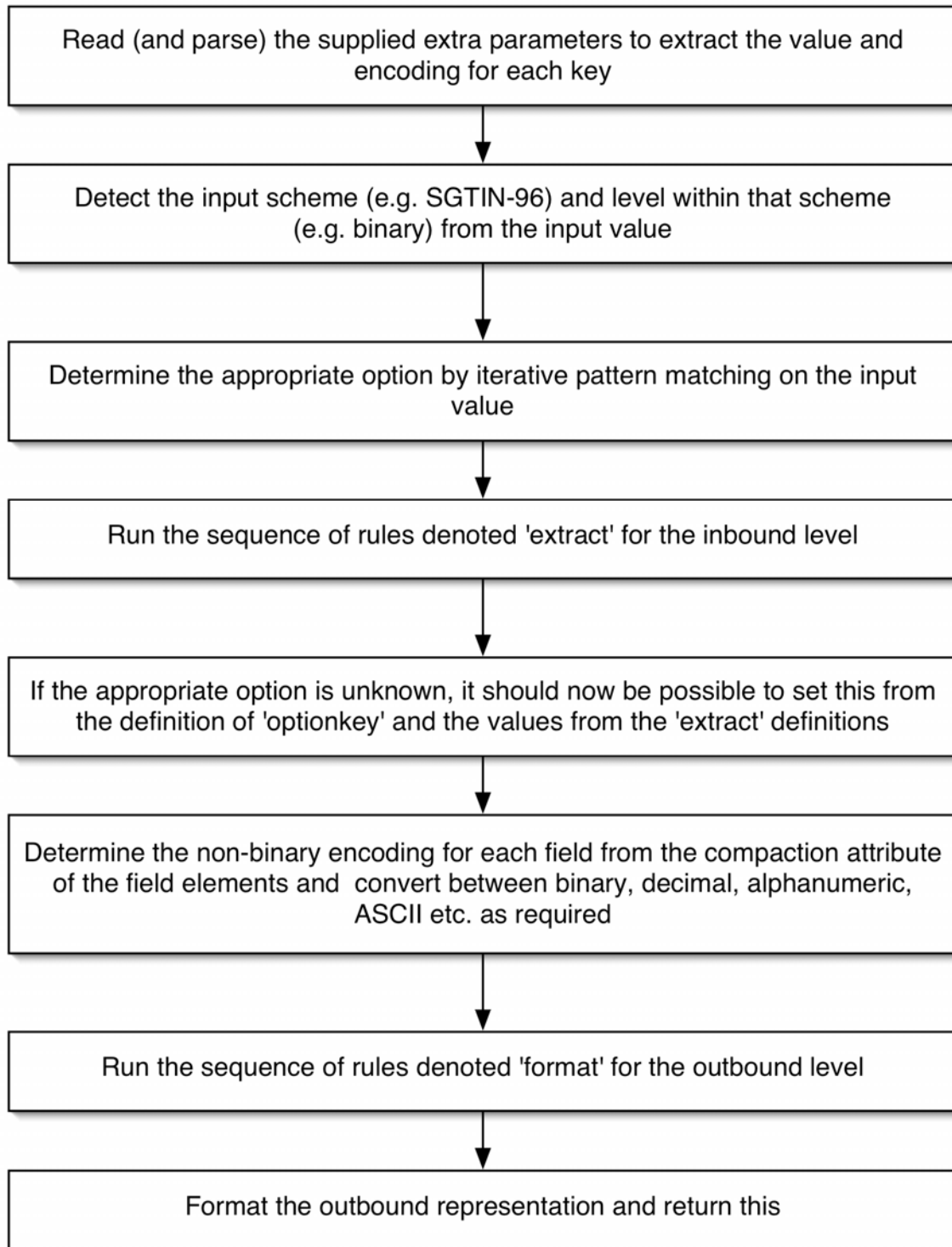
Stage	Description
EXTRACT	Operates on fields after parsing of the inbound value
FORMAT	Operates on fields in order to prepare additional fields required by the grammar for formatting the output value.

997

*Table 8 – The two stages for processing rules in Tag Data Translation*

998

999 The rules for each scheme are within the context of a particular level of representation.  
1000 The first block of rules, 'EXTRACT' are tied to the inbound representation level. The last  
1001 block of rules, 'FORMAT' is tied to the outbound representation level. Each block may  
1002 consist of zero or more rule elements. The rules within each block are executed in a  
1003 strict order, as specified by an ascending integer-based sequence number, indicated by the  
1004 attribute 'seq' of the rule element.



1005

1006

1007

*Figure 9 – Logical flowchart for a Tag Data Translation process*

1008

Figure 9 shows a flowchart indicating the key stages of operation of the translation process. These are:

1009

- 1010 1. Read the input value and supplied parameters. Read the outbound level.  
1011 2. Determine the coding scheme and inbound representation level  
1012 (This may require `tagLength` to be specified as a supplied parameter)  
1013 3. Determine the appropriate option value matching the input value string.  
1014 4. Read the regular expression for pattern matching  
1015 5. Populate any internal key-value lookup table or associative array with field values  
1016 extracted from parsing of the input value string using the regular expression and from  
1017 supplied parameters specified.  
1018 6. Read the grammar for the outbound representation and check whether the key-value  
1019 lookup table already contains all the field values needed for output. Supplied  
1020 parameters required for formatting are indicated via the  
1021 `requiredFormattingParameters` attribute of the `level` element.  
1022 7. Check whether the values of the fields are within the ranges allowed for the output  
1023 and throw errors if any fall outside the permitted numeric ranges, character sets or  
1024 string lengths.  
1025 8. If any of the field values are missing, parse the rules section of the TDT markup to  
1026 determine how to obtain the missing required fields in the key-value lookup table –  
1027 either by performing mathematical or string manipulations on other parameters, or  
1028 using lookups in other tables based on known fields as lookup keys.  
1029 9. Prepare the output string, using the key-value lookup table (which contains the values  
1030 of the fields), the output grammar expression and taking into account the length of  
1031 fields, and converting as necessary to/from binary.  
1032 10. Return the output string and any error messages – e.g. undefined fields, fields out of  
1033 numeric range, fields outside of character-set range

## 1034 **5.1. Tag Data Translation Software - Reference Implementation**

1035 A reference implementation may be a package / object class or subroutine, which may be  
1036 used at any part of the EPC Network technology stack and integrated with existing  
1037 software. Additionally, for educational and testing purposes, it will be useful to make a  
1038 Tag Data Translation capability available as a standalone service, with interaction either  
1039 via a web page form for a human operator or via a web service interface for automated  
1040 use, enabling efficient batch conversions.

## 1041 **6. Application Programming Interface**

1042 There are essentially two interfaces to consider for Tag Data Translation software,  
1043 namely a client-side interface, which provides conversion methods for users and a  
1044 maintenance interface, which ensures that the translation software is kept up to date with  
1045 the latest encoding/decoding definitions data.

1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082

## 6.1. Client API

```
public String translate(String epcIdentifier, String  
parameterList, String outputFormat)
```

Translates `epcIdentifier` from one representation into another within the same coding scheme.

### Parameters:

`epcIdentifier` – The `epcIdentifier` to be converted. This should be expressed as a string, in accordance with one of the grammars or patterns in the TDT markup files, i.e. a binary string consisting of characters '0' and '1', a URI (either tag-encoding or pure-identity formats), or a serialized legacy code expressed as in Table 3.

`parameterList` – This is a parameter string containing key value pairs, using the semicolon [';'] as delimiter between key=value pairs. For example, to convert a GTIN code the parameter string would look like the following:

`filter=3;companyprefixlength=7;taglength=96`

`outputFormat` – The output format into which the `epcIdentifier` SHALL be converted. The following are the formats supported:

1. BINARY
2. LEGACY
3. TAG\_ENCODING
4. PURE\_IDENTITY
5. ONS\_HOSTNAME

### Returns:

The converted value into one of the above formats as String.

### Throws:

TDTTranslationException – Throws exceptions due to the following reason:

1. `TDTFileNotFound` – Reports if the engine could not locate the configured definition file to compile.
2. `TDTFieldBelowMinimum` – Reports a (numeric) Field that fell below the `decimalMinimum` value allowed by the TDT markup
3. `TDTFieldAboveMaximum` – Reports a (numeric) Field that exceeded the `decimalMaximum` value allowed by the TDT markup

- 1083        4. TDTFieldOutsideCharacterSet - Reports a Field containing  
1084            characters outside the characterSet range allowed by the TDT markup
- 1085        5. TDTUndefinedField - Reports a Field required for the output or an  
1086            intermediate rule, whose value is undefined
- 1087        6. TDTSchemeNotFound - Reported if no matching Scheme can be found  
1088            via prefixMatch
- 1089        7. TDTLevelNotFound - Reported if no matching Level can be found via  
1090            prefixMatch
- 1091        8. TDTOptionNotFound - Reported if no matching Option can be found  
1092            via the optionKey or via matching the pattern
- 1093        9. TDTLookupFailed - Reported if lookup in an external table failed to  
1094            provide a value – reports table URI and path expression.
- 1095        10. TDTNumericOverflow - Reported when a numeric  
1096            overflow occurs when handling numeric values such  
1097            as serial number.

## 1098        **6.2. Maintenance API**

1099        public void **refreshTranslations()**

1100        Checks each subscription for any update, reloading new rules where necessary and forces  
1101        the software to reload or recompile its internal representation of the encoding/decoding  
1102        rules based on the current remaining subscriptions.



## 1103 7. TDT Schema Definition

1104

### 1105 EpcTagDataTranslation.xsd

```
1106 <?xml version="1.0" encoding="UTF-8"?>
1107 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1108 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1109 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1110 attributeFormDefault="unqualified" version="1.0">
1111   <xsd:annotation>
1112     <xsd:documentation><![CDATA[
1113       <epcglobal:copyright>Copyright ©2004 Epcglobal Inc., All Rights
1114       Reserved.</epcglobal:copyright>
1115       <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1116       employees, or agents shall not be liable for any injury, loss, damages,
1117       financial or otherwise, arising from, related to, or caused by the use of this
1118       document. The use of said document shall constitute your express consent to
1119       the foregoing exculpation.</epcglobal:disclaimer>
1120       <epcglobal:specification>Tag Data Translation (TDT) version
1121       1.0</epcglobal:specification>
1122     ]]></xsd:documentation>
1123   </xsd:annotation>
1124
1125   <xsd:include schemaLocation="LevelTypeList.xsd"/>
1126   <xsd:include schemaLocation="TagLengthList.xsd"/>
1127   <xsd:include schemaLocation="SchemeNameList.xsd"/>
1128   <xsd:include schemaLocation="InputFormatList.xsd"/>
1129   <xsd:include schemaLocation="ModeList.xsd"/>
1130   <xsd:include schemaLocation="CompactionMethodList.xsd"/>
1131   <xsd:include schemaLocation="PadDirectionList.xsd"/>
1132
1133   <xsd:complexType name="Field">
1134     <xsd:attribute name="seq" type="xsd:integer" use="required"/>
1135     <xsd:attribute name="name" type="xsd:string" use="required"/>
1136     <xsd:attribute name="bitLength" type="xsd:integer" />
1137     <xsd:attribute name="characterSet" type="xsd:string" use="required"/>
1138     <xsd:attribute name="compaction" type="tdt:CompactionMethodList"/>
1139     <xsd:attribute name="compression" type="xsd:string"/>
1140     <xsd:attribute name="padChar" type="xsd:string"/>
1141     <xsd:attribute name="padDir" type="tdt:PadDirectionList"/>
1142     <xsd:attribute name="decimalMinimum" type="xsd:long"/>
1143     <xsd:attribute name="decimalMaximum" type="xsd:long"/>
1144     <xsd:attribute name="length" type="xsd:integer"/>
1145   </xsd:complexType>
1146
1147   <xsd:complexType name="Option">
1148     <xsd:sequence>
1149       <xsd:element name="field" type="tdt:Field" maxOccurs="unbounded"/>
1150     </xsd:sequence>
1151     <xsd:attribute name="optionKey" type="xsd:string" use="required"/>
1152     <xsd:attribute name="pattern" type="xsd:string"/>
1153     <xsd:attribute name="grammar" type="xsd:string" use="required"/>
1154   </xsd:complexType>
1155
1156   <xsd:complexType name="Rule">
1157     <xsd:attribute name="type" type="tdt:ModeList" use="required"/>
1158     <xsd:attribute name="inputFormat" type="tdt:InputFormatList"
1159     use="required"/>
1160     <xsd:attribute name="seq" type="xsd:integer" use="required"/>
1161     <xsd:attribute name="newFieldName" type="xsd:string" use="required"/>
```

```

1162     <xsd:attribute name="characterSet" type="xsd:string" use="required"/>
1163     <xsd:attribute name="padChar" type="xsd:string"/>
1164     <xsd:attribute name="padDir" type="tdt:PadDirectionList"/>
1165     <xsd:attribute name="decimalMinimum" type="xsd:long"/>
1166     <xsd:attribute name="decimalMaximum" type="xsd:long"/>
1167     <xsd:attribute name="length" type="xsd:string"/>
1168     <xsd:attribute name="function" type="xsd:string" use="required"/>
1169     <xsd:attribute name="tableURL" type="xsd:string"/>
1170     <xsd:attribute name="tableParams" type="xsd:string"/>
1171     <xsd:attribute name="tableXPath" type="xsd:string"/>
1172     <xsd:attribute name="tableSQL" type="xsd:string"/>
1173 </xsd:complexType>
1174
1175 <xsd:complexType name="Level">
1176   <xsd:sequence>
1177     <xsd:element name="option" type="tdt:Option" minOccurs="1"
1178 maxOccurs="unbounded"/>
1179     <xsd:element name="rule" type="tdt:Rule" minOccurs="0"
1180 maxOccurs="unbounded"/>
1181   </xsd:sequence>
1182   <xsd:attribute name="type" type="tdt:LevelTypeList" use="required"/>
1183   <xsd:attribute name="prefixMatch" type="xsd:string" use="required"/>
1184   <xsd:attribute name="requiredParsingParameters" type="xsd:string"/>
1185   <xsd:attribute name="requiredFormattingParameters" type="xsd:string"/>
1186 </xsd:complexType>
1187
1188 <xsd:complexType name="Scheme">
1189   <xsd:sequence>
1190     <xsd:element name="level" type="tdt:Level" minOccurs="4" maxOccurs="5"/>
1191   </xsd:sequence>
1192   <xsd:attribute name="name" type="tdt:SchemeNameList" use="required"/>
1193   <xsd:attribute name="optionKey" type="xsd:string" use="required"/>
1194   <xsd:attribute name="tagLength" type="tdt:TagLengthList" use="required"/>
1195 </xsd:complexType>
1196
1197 <xsd:complexType name="EpcTagDataTranslation">
1198   <xsd:sequence>
1199     <xsd:element name="scheme" type="tdt:Scheme" maxOccurs="unbounded"/>
1200   </xsd:sequence>
1201   <xsd:attribute name="version" type="xsd:string" use="required"/>
1202   <xsd:attribute name="date" type="xsd:dateTime" use="required"/>
1203   <xsd:attribute name="epcTDSVersion" type="xsd:string" use="required"/>
1204 </xsd:complexType>
1205
1206 <xsd:element name="epcTagDataTranslation" type="tdt:EpcTagDataTranslation"/>
1207 </xsd:schema>
1208
1209

```

## 1210 LevelTypeList.xsd

```

1211 <?xml version="1.0" encoding="UTF-8"?>
1212 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1213 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1214 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1215 attributeFormDefault="unqualified" version="1.0">
1216   <xsd:annotation>
1217     <xsd:documentation><![CDATA[
1218       <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1219       Reserved.</epcglobal:copyright>
1220       <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1221       employees, or agents shall not be liable for any injury, loss, damages,
1222       financial or otherwise, arising from, related to, or caused by the use of this

```

```

1223 document. The use of said document shall constitute your express consent to
1224 the foregoing exculpation.</epcglobal:disclaimer>
1225 <epcglobal:specification>Tag Data Translation (TDT) version
1226 1.0</epcglobal:specification>
1227 ]]></xsd:documentation>
1228 </xsd:annotation>
1229 <xsd:simpleType name="LevelTypeList">
1230 <xsd:restriction base="xsd:string">
1231 <xsd:enumeration value="BINARY"/>
1232 <xsd:enumeration value="TAG_ENCODING"/>
1233 <xsd:enumeration value="PURE_IDENTITY"/>
1234 <xsd:enumeration value="LEGACY"/>
1235 <xsd:enumeration value="ONS_HOSTNAME"/>
1236 </xsd:restriction>
1237 </xsd:simpleType>
1238 </xsd:schema>
1239
1240

```

### 1241 TagLengthList.xsd

```

1242 <?xml version="1.0" encoding="UTF-8"?>
1243 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1244 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1245 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1246 attributeFormDefault="unqualified" version="1.0">
1247 <xsd:annotation>
1248 <xsd:documentation><![CDATA[
1249 <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1250 Reserved.</epcglobal:copyright>
1251 <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1252 employees, or agents shall not be liable for any injury, loss, damages,
1253 financial or otherwise, arising from, related to, or caused by the use of this
1254 document. The use of said document shall constitute your express consent to
1255 the foregoing exculpation.</epcglobal:disclaimer>
1256 <epcglobal:specification>Tag Data Translation (TDT) version
1257 1.0</epcglobal:specification>
1258 ]]></xsd:documentation>
1259 </xsd:annotation>
1260 <xsd:simpleType name="TagLengthList">
1261 <xsd:restriction base="xsd:string">
1262 <xsd:enumeration value="64"/>
1263 <xsd:enumeration value="96"/>
1264 </xsd:restriction>
1265 </xsd:simpleType>
1266 </xsd:schema>
1267
1268

```

### 1269 SchemeNameList.xsd

```

1270 <?xml version="1.0" encoding="UTF-8"?>
1271 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1272 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1273 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1274 attributeFormDefault="unqualified" version="1.0">
1275 <xsd:annotation>
1276 <xsd:documentation><![CDATA[
1277 <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1278 Reserved.</epcglobal:copyright>
1279 <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1280 employees, or agents shall not be liable for any injury, loss, damages,
1281 financial or otherwise, arising from, related to, or caused by the use of this

```

```

1282 document. The use of said document shall constitute your express consent to
1283 the foregoing exculpation.</epcglobal:disclaimer>
1284 <epcglobal:specification>Tag Data Translation (TDT) version
1285 1.0</epcglobal:specification>
1286 ]]></xsd:documentation>
1287 </xsd:annotation>
1288 <xsd:simpleType name="SchemeNameList">
1289 <xsd:restriction base="xsd:string">
1290 <xsd:enumeration value="SGTIN-64"/>
1291 <xsd:enumeration value="SGTIN-96"/>
1292 <xsd:enumeration value="SGLN-64"/>
1293 <xsd:enumeration value="SGLN-96"/>
1294 <xsd:enumeration value="GRAI-64"/>
1295 <xsd:enumeration value="GRAI-96"/>
1296 <xsd:enumeration value="GIAI-64"/>
1297 <xsd:enumeration value="GIAI-96"/>
1298 <xsd:enumeration value="SSCC-64"/>
1299 <xsd:enumeration value="SSCC-96"/>
1300 <xsd:enumeration value="GID-96"/>
1301 <xsd:enumeration value="USDOD-64"/>
1302 <xsd:enumeration value="USDOD-96"/>
1303 </xsd:restriction>
1304 </xsd:simpleType>
1305 </xsd:schema>
1306
1307

```

### 1308 InputFormatList.xsd

```

1309 <?xml version="1.0" encoding="UTF-8"?>
1310 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1311 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1312 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1313 attributeFormDefault="unqualified" version="1.0">
1314 <xsd:annotation>
1315 <xsd:documentation><![CDATA[
1316 <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1317 Reserved.</epcglobal:copyright>
1318 <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1319 employees, or agents shall not be liable for any injury, loss, damages,
1320 financial or otherwise, arising from, related to, or caused by the use of this
1321 document. The use of said document shall constitute your express consent to
1322 the foregoing exculpation.</epcglobal:disclaimer>
1323 <epcglobal:specification>Tag Data Translation (TDT) version
1324 1.0</epcglobal:specification>
1325 ]]></xsd:documentation>
1326 </xsd:annotation>
1327 <xsd:simpleType name="InputFormatList">
1328 <xsd:restriction base="xsd:string">
1329 <xsd:enumeration value="BINARY"/>
1330 <xsd:enumeration value="STRING"/>
1331 </xsd:restriction>
1332 </xsd:simpleType>
1333 </xsd:schema>
1334
1335

```

### 1336 ModeList.xsd

```

1337 <?xml version="1.0" encoding="UTF-8"?>
1338 <!-- Generated from annotated java -->
1339 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1340 xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

1341 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1342 attributeFormDefault="unqualified" version="1.0">
1343   <xsd:annotation>
1344     <xsd:documentation><![CDATA[
1345       <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1346       Reserved.</epcglobal:copyright>
1347       <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1348       employees, or agents shall not be liable for any injury, loss, damages,
1349       financial or otherwise, arising from, related to, or caused by the use of this
1350       document. The use of said document shall constitute your express consent to
1351       the foregoing exculpation.</epcglobal:disclaimer>
1352       <epcglobal:specification>Tag Data Translation (TDT) version
1353       1.0</epcglobal:specification>
1354     ]]></xsd:documentation>
1355   </xsd:annotation>
1356   <xsd:simpleType name="ModeList">
1357     <xsd:restriction base="xsd:string">
1358       <xsd:enumeration value="EXTRACT"/>
1359       <xsd:enumeration value="FORMAT"/>
1360     </xsd:restriction>
1361   </xsd:simpleType>
1362 </xsd:schema>
1363
1364

```

### 1365 CompactionMethodList.xsd

```

1366 <?xml version="1.0" encoding="UTF-8"?>
1367 <!-- Generated from annotated java -->
1368 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1369 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1370 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1371 attributeFormDefault="unqualified" version="1.0">
1372   <xsd:annotation>
1373     <xsd:documentation><![CDATA[
1374       <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1375       Reserved.</epcglobal:copyright>
1376       <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1377       employees, or agents shall not be liable for any injury, loss, damages,
1378       financial or otherwise, arising from, related to, or caused by the use of this
1379       document. The use of said document shall constitute your express consent to
1380       the foregoing exculpation.</epcglobal:disclaimer>
1381       <epcglobal:specification>Tag Data Translation (TDT) version
1382       1.0</epcglobal:specification>
1383     ]]></xsd:documentation>
1384   </xsd:annotation>
1385   <xsd:simpleType name="CompactionMethodList">
1386     <xsd:restriction base="xsd:string">
1387       <xsd:enumeration value="32-bit"/>
1388       <xsd:enumeration value="16-bit"/>
1389       <xsd:enumeration value="8-bit"/>
1390       <xsd:enumeration value="7-bit"/>
1391       <xsd:enumeration value="6-bit"/>
1392       <xsd:enumeration value="5-bit"/>
1393     </xsd:restriction>
1394   </xsd:simpleType>
1395 </xsd:schema>
1396

```

### 1397 PadDirectionList.xsd

```

1398 <?xml version="1.0" encoding="UTF-8"?>
1399 <!-- Generated from annotated java -->

```

```
1400 <xsd:schema targetNamespace="urn:epcglobal:tdt:xsd:1"
1401 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1402 xmlns:tdt="urn:epcglobal:tdt:xsd:1" elementFormDefault="unqualified"
1403 attributeFormDefault="unqualified" version="1.0">
1404   <xsd:annotation>
1405     <xsd:documentation><![CDATA[
1406       <epcglobal:copyright>Copyright ©2005 Epcglobal Inc., All Rights
1407       Reserved.</epcglobal:copyright>
1408       <epcglobal:disclaimer>EPCglobal Inc., its members, officers, directors,
1409       employees, or agents shall not be liable for any injury, loss, damages,
1410       financial or otherwise, arising from, related to, or caused by the use of this
1411       document. The use of said document shall constitute your express consent to
1412       the foregoing exculpation.</epcglobal:disclaimer>
1413       <epcglobal:specification>Tag Data Translation (TDT) version
1414       1.0</epcglobal:specification>
1415     ]]></xsd:documentation>
1416   </xsd:annotation>
1417   <xsd:simpleType name="PadDirectionList">
1418     <xsd:restriction base="xsd:string">
1419       <xsd:enumeration value="LEFT"/>
1420       <xsd:enumeration value="RIGHT"/>
1421     </xsd:restriction>
1422   </xsd:simpleType>
1423 </xsd:schema>
1424
1425
1426
1427
```

## 1428 8. TDT Markup V1.0

1429 The final draft Tag Data Translation markup files as defined in the TDS Version 1.1r1.27  
1430 will be made freely available at the following address:

1431 <http://www.autoidlabs.org/Cambridge/TDT>

1432 The markup files are also embedded in this appendix for ease of reference.

### 1433 8.1. GID-96 General Identifier

```
1434 <?xml version="1.0" encoding="UTF-8"?>
1435 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
1436 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
1437 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
1438 <scheme name="GID-96" optionKey="1" tagLength="96" >
1439 <level type="BINARY" prefixMatch="00110101" requiredFormattingParameters="taglength" >
1440 <option optionKey="1" pattern="00110101([01]{28})([01]{24})([01]{36})"
1441 grammar="'00110101' generalmanager objectclass serial" >
1442 <field seq="1" decimalMinimum="0" decimalMaximum="268435455" characterSet="[01]*"
1443 bitLength="28" name="generalmanager"/>
1444 <field seq="2" decimalMinimum="0" decimalMaximum="16777215" characterSet="[01]*"
1445 bitLength="24" name="objectclass"/>
1446 <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[01]*"
1447 bitLength="36" name="serial"/>
1448 </option>
1449 </level>
1450 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:gid-96"
1451 requiredFormattingParameters="taglength" >
1452 <option optionKey="1" pattern="urn:epc:tag:gid-96:([0-9]*)\.[0-9]*\.[0-9]*"
1453 grammar="'urn:epc:tag:gid-96:' generalmanager '.' objectclass '.' serial" >
1454 <field seq="1" decimalMinimum="0" decimalMaximum="268435455" characterSet="[0-9]*"
1455 name="generalmanager"/>
1456 <field seq="2" decimalMinimum="0" decimalMaximum="16777215" characterSet="[0-9]*"
1457 name="objectclass"/>
1458 <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
1459 name="serial"/>
1460 </option>
1461 </level>
1462 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:gid" >
1463 <option optionKey="1" pattern="urn:epc:id:gid:([0-9]*)\.[0-9]*\.[0-9]*"
1464 grammar="'urn:epc:id:gid:' generalmanager '.' objectclass '.' serial" >
1465 <field seq="1" decimalMinimum="0" decimalMaximum="268435455" characterSet="[0-9]*"
1466 name="generalmanager"/>
1467 <field seq="2" decimalMinimum="0" decimalMaximum="16777215" characterSet="[0-9]*"
1468 name="objectclass"/>
1469 <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
1470 name="serial"/>
1471 </option>
1472 </level>
1473 <level type="LEGACY" prefixMatch="generalmanager=" >
1474 <option optionKey="1" pattern="generalmanager=([0-9]*);objectclass=([0-
1475 9]*);serial=([0-9]*)" grammar="'generalmanager=' generalmanager ';objectclass='
1476 objectclass ';serial=' serial" >
1477 <field seq="1" decimalMinimum="0" decimalMaximum="268435455" characterSet="[0-9]*"
1478 name="generalmanager"/>
1479 <field seq="2" decimalMinimum="0" decimalMaximum="16777215" characterSet="[0-9]*"
1480 name="objectclass"/>
1481 <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
1482 name="serial"/>
1483 </option>
1484 </level>
1485 </scheme>
1486 </epcTagDataTranslation>
```

1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555

## 8.2. SGTIN-64 Serialized Global Trade Item Number for 64-bit EPCs

```
<?xml version="1.0" encoding="UTF-8"?>
<epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
xmlns:xsi="http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
  <scheme name="SGTIN-64" optionKey="companyprefixlength" tagLength="64">
    <level type="BINARY" prefixMatch="10" requiredFormattingParameters="filter,taglength" >
      <option optionKey="12" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
        <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
bitLength="14" name="companyprefixindex"/>
        <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[01]*"
bitLength="20" length="1" padChar="0" padDir="LEFT" name="itemref"/>
        <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]*"
bitLength="25" name="serial"/>
      </option>
      <option optionKey="11" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
        <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
bitLength="14" name="companyprefixindex"/>
        <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[01]*"
bitLength="20" length="2" padChar="0" padDir="LEFT" name="itemref"/>
        <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]*"
bitLength="25" name="serial"/>
      </option>
      <option optionKey="10" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
        <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
bitLength="14" name="companyprefixindex"/>
        <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[01]*"
bitLength="20" length="3" padChar="0" padDir="LEFT" name="itemref"/>
        <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]*"
bitLength="25" name="serial"/>
      </option>
      <option optionKey="9" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
        <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
bitLength="14" name="companyprefixindex"/>
        <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[01]*"
bitLength="20" length="4" padChar="0" padDir="LEFT" name="itemref"/>
        <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]*"
bitLength="25" name="serial"/>
      </option>
      <option optionKey="8" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
        <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
bitLength="14" name="companyprefixindex"/>
        <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
bitLength="20" length="5" padChar="0" padDir="LEFT" name="itemref"/>
        <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]*"
bitLength="25" name="serial"/>
      </option>
      <option optionKey="7" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
grammar="'10' filter companyprefixindex itemref serial" >
        <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
```



```

1556     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]"*
1557 bitLength="14" name="companyprefixindex"/>
1558     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]"*
1559 bitLength="20" length="6" padChar="0" padDir="LEFT" name="itemref"/>
1560     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]"*
1561 bitLength="25" name="serial"/>
1562 </option>
1563 <option optionKey="6" pattern="10([01]{3})([01]{14})([01]{20})([01]{25})"
1564 grammar="'10' filter companyprefixindex itemref serial" >
1565     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]"*
1566 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1567     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]"*
1568 bitLength="14" name="companyprefixindex"/>
1569     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]"*
1570 bitLength="20" length="7" padChar="0" padDir="LEFT" name="itemref"/>
1571     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[01]"*
1572 bitLength="25" name="serial"/>
1573 </option>
1574 <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="companyprefix"
1575 characterSet="[0-9]"*
1576 function="TABLELOOKUP(companyprefixindex,tdt64bitcpi,companyprefixindex,companyprefix)"
1577 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
1578 tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"
1579 tableParams="companyprefixindex"/>
1580 <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefixlength"
1581 characterSet="[0-9]"* function="LENGTH(companyprefix)"/>
1582 <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="companyprefixindex"
1583 characterSet="[0-9]"*
1584 function="TABLELOOKUP(companyprefix,tdt64bitcpi,companyprefix,companyprefixindex)"
1585 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
1586 tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index" tableParams="companyprefix"/>
1587 </level>
1588 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sgtin-64"
1589 requiredFormattingParameters="filter,taglength" >
1590 <option optionKey="12" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.\([0-9]{12})\.\([0-
1591 9]{1})\.\([0-9]*)" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1592 '.' serial" >
1593     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1594 length="1" padChar="0" padDir="LEFT" name="filter"/>
1595     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-
1596 9]"* length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
1597     <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]"*
1598 length="1" padChar="0" padDir="LEFT" name="itemref"/>
1599     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]"*
1600 name="serial"/>
1601 </option>
1602 <option optionKey="11" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.\([0-9]{11})\.\([0-
1603 9]{2})\.\([0-9]*)" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1604 '.' serial" >
1605     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1606 length="1" padChar="0" padDir="LEFT" name="filter"/>
1607     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]"*
1608 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
1609     <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]"*
1610 length="2" padChar="0" padDir="LEFT" name="itemref"/>
1611     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]"*
1612 name="serial"/>
1613 </option>
1614 <option optionKey="10" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.\([0-9]{10})\.\([0-
1615 9]{3})\.\([0-9]*)" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1616 '.' serial" >
1617     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1618 length="1" padChar="0" padDir="LEFT" name="filter"/>
1619     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]"*
1620 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1621     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]"*
1622 length="3" padChar="0" padDir="LEFT" name="itemref"/>
1623     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]"*
1624 name="serial"/>
1625 </option>

```

```

1626     <option optionKey="9" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.[(0-9]{9})\.[(0-
1627 9]{4})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1628 '.' serial" >
1629     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1630 length="1" padChar="0" padDir="LEFT" name="filter"/>
1631     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
1632 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
1633     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
1634 length="4" padChar="0" padDir="LEFT" name="itemref"/>
1635     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1636 name="serial"/>
1637 </option>
1638 <option optionKey="8" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.[(0-9]{8})\.[(0-
1639 9]{5})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1640 '.' serial" >
1641     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1642 length="1" padChar="0" padDir="LEFT" name="filter"/>
1643     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
1644 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
1645     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
1646 length="5" padChar="0" padDir="LEFT" name="itemref"/>
1647     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1648 name="serial"/>
1649 </option>
1650 <option optionKey="7" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.[(0-9]{7})\.[(0-
1651 9]{6})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1652 '.' serial" >
1653     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1654 length="1" padChar="0" padDir="LEFT" name="filter"/>
1655     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
1656 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
1657     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
1658 length="6" padChar="0" padDir="LEFT" name="itemref"/>
1659     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1660 name="serial"/>
1661 </option>
1662 <option optionKey="6" pattern="urn:epc:tag:sgtin-64:([0-7]{1})\.[(0-9]{6})\.[(0-
1663 9]{7})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-64:' filter '.' companyprefix '.' itemref
1664 '.' serial" >
1665     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1666 length="1" padChar="0" padDir="LEFT" name="filter"/>
1667     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
1668 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
1669     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[0-9]*"
1670 length="7" padChar="0" padDir="LEFT" name="itemref"/>
1671     <field seq="4" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1672 name="serial"/>
1673 </option>
1674 </level>
1675 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sgtin" >
1676     <option optionKey="12" pattern="urn:epc:id:sgtin:([0-9]{12})\.[(0-9]{1})\.[(0-9)*"
1677 grammar="urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1678     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
1679 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
1680     <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
1681 length="1" padChar="0" padDir="LEFT" name="itemref"/>
1682     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1683 name="serial"/>
1684 </option>
1685 <option optionKey="11" pattern="urn:epc:id:sgtin:([0-9]{11})\.[(0-9]{2})\.[(0-9)*"
1686 grammar="urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1687     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-9]*"
1688 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
1689     <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
1690 length="2" padChar="0" padDir="LEFT" name="itemref"/>
1691     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1692 name="serial"/>
1693 </option>
1694 <option optionKey="10" pattern="urn:epc:id:sgtin:([0-9]{10})\.[(0-9]{3})\.[(0-9)*"
1695 grammar="urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >

```

```

1696     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
1697 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1698     <field seq="2" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
1699 length="3" padChar="0" padDir="LEFT" name="itemref"/>
1700     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1701 name="serial"/>
1702 </option>
1703     <option optionKey="9" pattern="urn:epc:id:sgtin:([0-9]{9})\.([0-9]{4})\.([0-9]*)"
1704 grammar="'urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1705     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
1706 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
1707     <field seq="2" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
1708 length="4" padChar="0" padDir="LEFT" name="itemref"/>
1709     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1710 name="serial"/>
1711 </option>
1712     <option optionKey="8" pattern="urn:epc:id:sgtin:([0-9]{8})\.([0-9]{5})\.([0-9]*)"
1713 grammar="'urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1714     <field seq="1" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
1715 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
1716     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
1717 length="5" padChar="0" padDir="LEFT" name="itemref"/>
1718     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1719 name="serial"/>
1720 </option>
1721     <option optionKey="7" pattern="urn:epc:id:sgtin:([0-9]{7})\.([0-9]{6})\.([0-9]*)"
1722 grammar="'urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1723     <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
1724 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
1725     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
1726 length="6" padChar="0" padDir="LEFT" name="itemref"/>
1727     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1728 name="serial"/>
1729 </option>
1730     <option optionKey="6" pattern="urn:epc:id:sgtin:([0-9]{6})\.([0-9]{7})\.([0-9]*)"
1731 grammar="'urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
1732     <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
1733 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
1734     <field seq="2" decimalMinimum="0" decimalMaximum="1048575" characterSet="[0-9]*"
1735 length="7" padChar="0" padDir="LEFT" name="itemref"/>
1736     <field seq="3" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1737 name="serial"/>
1738 </option>
1739 </level>
1740 <level type="LEGACY" prefixMatch="gtin="
1741 requiredParsingParameters="companyprefixlength" >
1742     <option optionKey="12" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
1743 indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
1744     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1745 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1746     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1747 name="serial"/>
1748 </option>
1749     <option optionKey="11" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
1750 indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
1751     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1752 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1753     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1754 name="serial"/>
1755 </option>
1756     <option optionKey="10" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
1757 indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
1758     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1759 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1760     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1761 name="serial"/>
1762 </option>
1763     <option optionKey="9" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
1764 indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
1765     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1766 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>

```

```

1767     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1768     name="serial"/>
1769   </option>
1770   <option optionKey="8" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin=
1771   indicatordigit companyprefix itemrefremainder checkdigit ;serial=' serial" >
1772     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1773   9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1774     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1775     name="serial"/>
1776   </option>
1777   <option optionKey="7" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin=
1778   indicatordigit companyprefix itemrefremainder checkdigit ;serial=' serial" >
1779     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1780   9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1781     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1782     name="serial"/>
1783   </option>
1784   <option optionKey="6" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin=
1785   indicatordigit companyprefix itemrefremainder checkdigit ;serial=' serial" >
1786     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1787   9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
1788     <field seq="2" decimalMinimum="0" decimalMaximum="33554431" characterSet="[0-9]*"
1789     name="serial"/>
1790   </option>
1791   <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="gtinprefixremainder"
1792   characterSet="[0-9]*" length="12" function="SUBSTR(gtin,1,12)"/>
1793   <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="indicatordigit"
1794   characterSet="[0-9]*" length="1" function="SUBSTR(gtin,0,1)"/>
1795   <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="itemrefremainder"
1796   characterSet="[0-9]*" function="SUBSTR(gtinprefixremainder,companyprefixlength)"/>
1797   <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="itemref"
1798   characterSet="[0-9]*" function="CONCAT(indicatordigit,itemrefremainder)"/>
1799   <rule type="EXTRACT" inputFormat="STRING" seq="5" newFieldName="companyprefix"
1800   characterSet="[0-9]*" function="SUBSTR(gtinprefixremainder,0,companyprefixlength)"/>
1801   <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="indicatordigit"
1802   characterSet="[0-9]*" length="1" function="SUBSTR(itemref,0,1)"/>
1803   <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="itemrefremainder"
1804   characterSet="[0-9]*" function="SUBSTR(itemref,1)"/>
1805   <rule type="FORMAT" inputFormat="STRING" seq="3" newFieldName="gtinprefix"
1806   characterSet="[0-9]*" length="13"
1807   function="CONCAT(indicatordigit,companyprefix,itemrefremainder)"/>
1808   <rule type="FORMAT" inputFormat="STRING" seq="4" newFieldName="checkdigit"
1809   characterSet="[0-9]*" length="1" function="GS1CHECKSUM(gtinprefix)"/>
1810 </level>
1811 <level type="ONS_HOSTNAME" >
1812   <option optionKey="12" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1813     <field seq="1" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
1814     length="1" padChar="0" padDir="LEFT" name="itemref"/>
1815     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
1816   9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
1817   </option>
1818   <option optionKey="11" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1819     <field seq="1" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
1820     length="2" padChar="0" padDir="LEFT" name="itemref"/>
1821     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-9]*"
1822     length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
1823   </option>
1824   <option optionKey="10" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1825     <field seq="1" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
1826     length="3" padChar="0" padDir="LEFT" name="itemref"/>
1827     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-9]*"
1828     length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1829   </option>
1830   <option optionKey="9" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1831     <field seq="1" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
1832     length="4" padChar="0" padDir="LEFT" name="itemref"/>
1833     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-9]*"
1834     length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
1835   </option>
1836   <option optionKey="8" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >

```

```

1837     <field seq="1" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
1838     length="5" padChar="0" padDir="LEFT" name="itemref"/>
1839     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
1840     length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
1841     </option>
1842     <option optionKey="7" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1843     <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
1844     length="6" padChar="0" padDir="LEFT" name="itemref"/>
1845     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
1846     length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
1847     </option>
1848     <option optionKey="6" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
1849     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
1850     length="7" padChar="0" padDir="LEFT" name="itemref"/>
1851     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
1852     length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
1853     </option>
1854     </level>
1855     </scheme>
1856 </epcTagDataTranslation>

```

### 1857 8.3. SGTIN-96 Serialized Global Trade Item Number for 96-bit 1858 EPCs

```

1859 <?xml version="1.0" encoding="UTF-8"?>
1860 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
1861 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
1862 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
1863   <scheme name="SGTIN-96" optionKey="companyprefixlength" tagLength="96" >
1864     <level type="BINARY" prefixMatch="00110000"
1865     requiredFormattingParameters="filter,taglength" >
1866       <option optionKey="12" pattern="00110000([01]{3})000([01]{40})([01]{4})([01]{38})"
1867       grammar="'00110000' filter '000' companyprefix itemref serial" >
1868         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1869         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1870         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
1871         bitLength="40" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
1872         <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[01]*"
1873         bitLength="4" length="1" padChar="0" padDir="LEFT" name="itemref"/>
1874         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1875         bitLength="38" name="serial"/>
1876       </option>
1877       <option optionKey="11" pattern="00110000([01]{3})001([01]{37})([01]{7})([01]{38})"
1878       grammar="'00110000' filter '001' companyprefix itemref serial" >
1879         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1880         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1881         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
1882         bitLength="37" length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
1883         <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[01]*"
1884         bitLength="7" length="2" padChar="0" padDir="LEFT" name="itemref"/>
1885         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1886         bitLength="38" name="serial"/>
1887       </option>
1888       <option optionKey="10" pattern="00110000([01]{3})010([01]{34})([01]{10})([01]{38})"
1889       grammar="'00110000' filter '010' companyprefix itemref serial" >
1890         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1891         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1892         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
1893         bitLength="34" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1894         <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[01]*"
1895         bitLength="10" length="3" padChar="0" padDir="LEFT" name="itemref"/>
1896         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1897         bitLength="38" name="serial"/>
1898       </option>
1899       <option optionKey="9" pattern="00110000([01]{3})011([01]{30})([01]{14})([01]{38})"
1900       grammar="'00110000' filter '011' companyprefix itemref serial" >
1901         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1902         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1903

```

```

1904     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
1905 bitLength="30" length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
1906     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[01]*"
1907 bitLength="14" length="4" padChar="0" padDir="LEFT" name="itemref"/>
1908     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1909 bitLength="38" name="serial"/>
1910 </option>
1911 <option optionKey="8" pattern="00110000([01]{3})100([01]{27})([01]{17})([01]{38})"
1912 grammar="'00110000' filter '100' companyprefix itemref serial" >
1913     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1914 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1915     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
1916 bitLength="27" length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
1917     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
1918 bitLength="17" length="5" padChar="0" padDir="LEFT" name="itemref"/>
1919     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1920 bitLength="38" name="serial"/>
1921 </option>
1922 <option optionKey="7" pattern="00110000([01]{3})101([01]{24})([01]{20})([01]{38})"
1923 grammar="'00110000' filter '101' companyprefix itemref serial" >
1924     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1925 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1926     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
1927 bitLength="24" length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
1928     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
1929 bitLength="20" length="6" padChar="0" padDir="LEFT" name="itemref"/>
1930     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1931 bitLength="38" name="serial"/>
1932 </option>
1933 <option optionKey="6" pattern="00110000([01]{3})110([01]{20})([01]{24})([01]{38})"
1934 grammar="'00110000' filter '110' companyprefix itemref serial" >
1935     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
1936 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
1937     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
1938 bitLength="20" length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
1939     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
1940 bitLength="24" length="7" padChar="0" padDir="LEFT" name="itemref"/>
1941     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
1942 bitLength="38" name="serial"/>
1943 </option>
1944 </level>
1945 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sgtin-96"
1946 requiredFormattingParameters="filter,taglength" >
1947     <option optionKey="12" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[0-9]{12}\.[0-9]{1}\.[0-9]*" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
1948 '.' serial" >
1949     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1950 length="1" padChar="0" padDir="LEFT" name="filter"/>
1951     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
1952     <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
1953 length="1" padChar="0" padDir="LEFT" name="itemref"/>
1954     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-9]*" name="serial"/>
1955 </option>
1956 <option optionKey="11" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[0-9]{11}\.[0-9]{2}\.[0-9]*" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
1957 '.' serial" >
1958     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1959 length="1" padChar="0" padDir="LEFT" name="filter"/>
1960     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
1961 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
1962     <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
1963 length="2" padChar="0" padDir="LEFT" name="itemref"/>
1964     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-9]*" name="serial"/>
1965 </option>
1966 <option optionKey="10" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[0-9]{10}\.[0-9]{3}\.[0-9]*" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
1967 '.' serial" >
1968     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
1969 length="1" padChar="0" padDir="LEFT" name="filter"/>
1970     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1971     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*" length="3" padChar="0" padDir="LEFT" name="itemref"/>
1972     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-9]*" name="serial"/>
1973 </option>

```

```

1974     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1975 length="1" padChar="0" padDir="LEFT" name="filter"/>
1976     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]"*
1977 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
1978     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]"*
1979 length="3" padChar="0" padDir="LEFT" name="itemref"/>
1980     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
1981 9]"* name="serial"/>
1982     </option>
1983     <option optionKey="9" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[(0-9){9}]\.[(0-
1984 9]{4})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
1985 '.' serial" >
1986     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1987 length="1" padChar="0" padDir="LEFT" name="filter"/>
1988     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]"*
1989 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
1990     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]"*
1991 length="4" padChar="0" padDir="LEFT" name="itemref"/>
1992     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
1993 9]"* name="serial"/>
1994     </option>
1995     <option optionKey="8" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[(0-9){8}]\.[(0-
1996 9]{5})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
1997 '.' serial" >
1998     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
1999 length="1" padChar="0" padDir="LEFT" name="filter"/>
2000     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]"*
2001 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2002     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]"*
2003 length="5" padChar="0" padDir="LEFT" name="itemref"/>
2004     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2005 9]"* name="serial"/>
2006     </option>
2007     <option optionKey="7" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[(0-9){7}]\.[(0-
2008 9]{6})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
2009 '.' serial" >
2010     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
2011 length="1" padChar="0" padDir="LEFT" name="filter"/>
2012     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]"*
2013 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2014     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]"*
2015 length="6" padChar="0" padDir="LEFT" name="itemref"/>
2016     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2017 9]"* name="serial"/>
2018     </option>
2019     <option optionKey="6" pattern="urn:epc:tag:sgtin-96:([0-7]{1})\.[(0-9){6}]\.[(0-
2020 9]{7})\.[(0-9)*]" grammar="urn:epc:tag:sgtin-96:' filter '.' companyprefix '.' itemref
2021 '.' serial" >
2022     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]"*
2023 length="1" padChar="0" padDir="LEFT" name="filter"/>
2024     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]"*
2025 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2026     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]"*
2027 length="7" padChar="0" padDir="LEFT" name="itemref"/>
2028     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2029 9]"* name="serial"/>
2030     </option>
2031     </level>
2032     <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sgtin" >
2033     <option optionKey="12" pattern="urn:epc:id:sgtin:([0-9]{12})\.[(0-9){1}]\.[(0-9)*]"
2034 grammar="urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >
2035     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2036 9]"* length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2037     <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]"*
2038 length="1" padChar="0" padDir="LEFT" name="itemref"/>
2039     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2040 9]"* name="serial"/>
2041     </option>
2042     <option optionKey="11" pattern="urn:epc:id:sgtin:([0-9]{11})\.[(0-9){2}]\.[(0-9)*]"
2043 grammar="urn:epc:id:sgtin:' companyprefix '.' itemref '.' serial" >

```

```

2044     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2045     length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2046     <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
2047     length="2" padChar="0" padDir="LEFT" name="itemref"/>
2048     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2049     9]*" name="serial"/>
2050     </option>
2051     <option optionKey="10" pattern="urn:epc:id:sgtin:([0-9]{10})\.[([0-9]{3})\.[([0-9]*)"
2052     grammar="urn:epc:id:sgtin: companyprefix '.' itemref '.' serial" >
2053     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2054     length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2055     <field seq="2" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
2056     length="3" padChar="0" padDir="LEFT" name="itemref"/>
2057     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2058     9]*" name="serial"/>
2059     </option>
2060     <option optionKey="9" pattern="urn:epc:id:sgtin:([0-9]{9})\.[([0-9]{4})\.[([0-9]*)"
2061     grammar="urn:epc:id:sgtin: companyprefix '.' itemref '.' serial" >
2062     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2063     length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2064     <field seq="2" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
2065     length="4" padChar="0" padDir="LEFT" name="itemref"/>
2066     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2067     9]*" name="serial"/>
2068     </option>
2069     <option optionKey="8" pattern="urn:epc:id:sgtin:([0-9]{8})\.[([0-9]{5})\.[([0-9]*)"
2070     grammar="urn:epc:id:sgtin: companyprefix '.' itemref '.' serial" >
2071     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2072     length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2073     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2074     length="5" padChar="0" padDir="LEFT" name="itemref"/>
2075     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2076     9]*" name="serial"/>
2077     </option>
2078     <option optionKey="7" pattern="urn:epc:id:sgtin:([0-9]{7})\.[([0-9]{6})\.[([0-9]*)"
2079     grammar="urn:epc:id:sgtin: companyprefix '.' itemref '.' serial" >
2080     <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2081     length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2082     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2083     length="6" padChar="0" padDir="LEFT" name="itemref"/>
2084     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2085     9]*" name="serial"/>
2086     </option>
2087     <option optionKey="6" pattern="urn:epc:id:sgtin:([0-9]{6})\.[([0-9]{7})\.[([0-9]*)"
2088     grammar="urn:epc:id:sgtin: companyprefix '.' itemref '.' serial" >
2089     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2090     length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2091     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2092     length="7" padChar="0" padDir="LEFT" name="itemref"/>
2093     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2094     9]*" name="serial"/>
2095     </option>
2096     </level>
2097     <level type="LEGACY" prefixMatch="gtin="
2098     requiredParsingParameters="companyprefixlength" >
2099     <option optionKey="12" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin='
2100     indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
2101     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999" characterSet="[0-
2102     9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2103     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2104     9]*" name="serial"/>
2105     </option>
2106     <option optionKey="11" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin='
2107     indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >
2108     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999999999" characterSet="[0-
2109     9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2110     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2111     9]*" name="serial"/>
2112     </option>
2113     <option optionKey="10" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar='gtin='
2114     indicatordigit companyprefix itemrefremainder checkdigit ';serial=' serial" >

```



```

2115     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
2116 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2117     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2118 9]*" name="serial"/>
2119     </option>
2120     <option optionKey="9" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
2121 indicatordigit companyprefix itemrefremainder checkdigit ' ;serial=' serial" >
2122     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
2123 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2124     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2125 9]*" name="serial"/>
2126     </option>
2127     <option optionKey="8" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
2128 indicatordigit companyprefix itemrefremainder checkdigit ' ;serial=' serial" >
2129     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
2130 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2131     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2132 9]*" name="serial"/>
2133     </option>
2134     <option optionKey="7" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
2135 indicatordigit companyprefix itemrefremainder checkdigit ' ;serial=' serial" >
2136     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
2137 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2138     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2139 9]*" name="serial"/>
2140     </option>
2141     <option optionKey="6" pattern="gtin=([0-9]{14});serial=([0-9]*)" grammar="'gtin='
2142 indicatordigit companyprefix itemrefremainder checkdigit ' ;serial=' serial" >
2143     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
2144 9]*" length="14" padChar="0" padDir="LEFT" name="gtin"/>
2145     <field seq="2" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
2146 9]*" name="serial"/>
2147     </option>
2148     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="gtinprefixremainder"
2149 characterSet="[0-9]*" length="12" function="SUBSTR(gtin,1,12)"/>
2150     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="indicatordigit"
2151 characterSet="[0-9]*" length="1" function="SUBSTR(gtin,0,1)"/>
2152     <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="itemrefremainder"
2153 characterSet="[0-9]*" function="SUBSTR(gtinprefixremainder,companyprefixlength)"/>
2154     <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="itemref"
2155 characterSet="[0-9]*" function="CONCAT(indicatordigit,itemrefremainder)"/>
2156     <rule type="EXTRACT" inputFormat="STRING" seq="5" newFieldName="companyprefix"
2157 characterSet="[0-9]*" function="SUBSTR(gtinprefixremainder,0,companyprefixlength)"/>
2158     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="indicatordigit"
2159 characterSet="[0-9]*" length="1" function="SUBSTR(itemref,0,1)"/>
2160     <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="itemrefremainder"
2161 characterSet="[0-9]*" function="SUBSTR(itemref,1)"/>
2162     <rule type="FORMAT" inputFormat="STRING" seq="3" newFieldName="gtinprefix"
2163 characterSet="[0-9]*" length="13"
2164 function="CONCAT(indicatordigit,companyprefix,itemrefremainder)"/>
2165     <rule type="FORMAT" inputFormat="STRING" seq="4" newFieldName="checkdigit"
2166 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(gtinprefix)"/>
2167     </level>
2168     <level type="ONS_HOSTNAME" >
2169     <option optionKey="12" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2170     <field seq="1" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
2171 length="1" padChar="0" padDir="LEFT" name="itemref"/>
2172     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2173 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2174     </option>
2175     <option optionKey="11" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2176     <field seq="1" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
2177 length="2" padChar="0" padDir="LEFT" name="itemref"/>
2178     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2179 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2180     </option>
2181     <option optionKey="10" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2182     <field seq="1" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
2183 length="3" padChar="0" padDir="LEFT" name="itemref"/>
2184     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2185 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>

```

```

2186     </option>
2187     <option optionKey="9" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2188       <field seq="1" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
2189       length="4" padChar="0" padDir="LEFT" name="itemref"/>
2190       <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2191       length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2192     </option>
2193     <option optionKey="8" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2194       <field seq="1" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2195       length="5" padChar="0" padDir="LEFT" name="itemref"/>
2196       <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2197       length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2198     </option>
2199     <option optionKey="7" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2200       <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2201       length="6" padChar="0" padDir="LEFT" name="itemref"/>
2202       <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2203       length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2204     </option>
2205     <option optionKey="6" grammar="itemref '.' companyprefix '.sgtin.id.onsepc.com'" >
2206       <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2207       length="7" padChar="0" padDir="LEFT" name="itemref"/>
2208       <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2209       length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2210     </option>
2211   </level>
2212 </scheme>
2213 </epcTagDataTranslation>

```

## 2214 8.4. SSCC-64 Serial Shipping Container Code for 64-bit EPCs

```

2215 <?xml version="1.0" encoding="UTF-8"?>
2216 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
2217 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
2218 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
2219   <scheme name="SSCC-64" optionKey="companyprefixlength" tagLength="64" >
2220     <level type="BINARY" prefixMatch="00001000"
2221     requiredFormattingParameters="filter,taglength" >
2222       <option optionKey="12" pattern="00001000([01]{3})([01]{14})([01]{39})"
2223       grammar="'00001000' filter companyprefixindex serialref" >
2224         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2225         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2226         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2227         bitLength="14" name="companyprefixindex"/>
2228         <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
2229         bitLength="39" length="5" padChar="0" padDir="LEFT" name="serialref"/>
2230       </option>
2231       <option optionKey="11" pattern="00001000([01]{3})([01]{14})([01]{39})"
2232       grammar="'00001000' filter companyprefixindex serialref" >
2233         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2234         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2235         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2236         bitLength="14" name="companyprefixindex"/>
2237         <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
2238         bitLength="39" length="6" padChar="0" padDir="LEFT" name="serialref"/>
2239       </option>
2240       <option optionKey="10" pattern="00001000([01]{3})([01]{14})([01]{39})"
2241       grammar="'00001000' filter companyprefixindex serialref" >
2242         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2243         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2244         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2245         bitLength="14" name="companyprefixindex"/>
2246         <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
2247         bitLength="39" length="7" padChar="0" padDir="LEFT" name="serialref"/>
2248       </option>
2249       <option optionKey="9" pattern="00001000([01]{3})([01]{14})([01]{39})"
2250       grammar="'00001000' filter companyprefixindex serialref" >
2251         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2252         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2253

```

```

2254     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2255 bitLength="14" name="companyprefixindex"/>
2256     <field seq="3" decimalMinimum="0" decimalMaximum="99999999" characterSet="[01]*"
2257 bitLength="39" length="8" padChar="0" padDir="LEFT" name="serialref"/>
2258     </option>
2259     <option optionKey="8" pattern="00001000([01]{3})([01]{14})([01]{39})"
2260 grammar="'00001000' filter companyprefixindex serialref" >
2261     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2262 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2263     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2264 bitLength="14" name="companyprefixindex"/>
2265     <field seq="3" decimalMinimum="0" decimalMaximum="99999999" characterSet="[01]*"
2266 bitLength="39" length="9" padChar="0" padDir="LEFT" name="serialref"/>
2267     </option>
2268     <option optionKey="7" pattern="00001000([01]{3})([01]{14})([01]{39})"
2269 grammar="'00001000' filter companyprefixindex serialref" >
2270     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2271 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2272     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2273 bitLength="14" name="companyprefixindex"/>
2274     <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
2275 bitLength="39" length="10" padChar="0" padDir="LEFT" name="serialref"/>
2276     </option>
2277     <option optionKey="6" pattern="00001000([01]{3})([01]{14})([01]{39})"
2278 grammar="'00001000' filter companyprefixindex serialref" >
2279     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2280 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2281     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2282 bitLength="14" name="companyprefixindex"/>
2283     <field seq="3" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
2284 bitLength="39" length="11" padChar="0" padDir="LEFT" name="serialref"/>
2285     </option>
2286     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="companyprefix"
2287 characterSet="[0-9]*"
2288 function="TABLELOOKUP(companyprefixindex,tdt64bitcpi,companyprefixindex,companyprefix)"
2289 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
2290 tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"
2291 tableParams="companyprefixindex"/>
2292     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefixlength"
2293 characterSet="[0-9]*" function="LENGTH(companyprefix)"/>
2294     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="companyprefixindex"
2295 characterSet="[0-9]*"
2296 function="TABLELOOKUP(companyprefix,tdt64bitcpi,companyprefix,companyprefixindex)"
2297 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
2298 tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index" tableParams="companyprefix"/>
2299     </level>
2300     <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sscc-64"
2301 requiredFormattingParameters="filter,taglength" >
2302     <option optionKey="12" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[(0-9){12}]\.[(0-
2303 9]{5})" grammar="'urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2304     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2305 length="1" padChar="0" padDir="LEFT" name="filter"/>
2306     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-
2307 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2308     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2309 length="5" padChar="0" padDir="LEFT" name="serialref"/>
2310     </option>
2311     <option optionKey="11" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[(0-9){11}]\.[(0-
2312 9){6})" grammar="'urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2313     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2314 length="1" padChar="0" padDir="LEFT" name="filter"/>
2315     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2316 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2317     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2318 length="6" padChar="0" padDir="LEFT" name="serialref"/>
2319     </option>
2320     <option optionKey="10" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[(0-9){10}]\.[(0-
2321 9){7})" grammar="'urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2322     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2323 length="1" padChar="0" padDir="LEFT" name="filter"/>

```

```

2324     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2325 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2326     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2327 length="7" padChar="0" padDir="LEFT" name="serialref"/>
2328     </option>
2329     <option optionKey="9" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[0-9]{9}\.[0-
2330 9]{8}" grammar="urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2331     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2332 length="1" padChar="0" padDir="LEFT" name="filter"/>
2333     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2334 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2335     <field seq="3" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2336 length="8" padChar="0" padDir="LEFT" name="serialref"/>
2337     </option>
2338     <option optionKey="8" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[0-9]{8}\.[0-
2339 9]{9}" grammar="urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2340     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2341 length="1" padChar="0" padDir="LEFT" name="filter"/>
2342     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2343 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2344     <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2345 length="9" padChar="0" padDir="LEFT" name="serialref"/>
2346     </option>
2347     <option optionKey="7" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[0-9]{7}\.[0-
2348 9]{10}" grammar="urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2349     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2350 length="1" padChar="0" padDir="LEFT" name="filter"/>
2351     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2352 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2353     <field seq="3" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2354 length="10" padChar="0" padDir="LEFT" name="serialref"/>
2355     </option>
2356     <option optionKey="6" pattern="urn:epc:tag:sscc-64:([0-7]{1})\.[0-9]{6}\.[0-
2357 9]{11}" grammar="urn:epc:tag:sscc-64:' filter '.' companyprefix '.' serialref" >
2358     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2359 length="1" padChar="0" padDir="LEFT" name="filter"/>
2360     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2361 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2362     <field seq="3" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2363 length="11" padChar="0" padDir="LEFT" name="serialref"/>
2364     </option>
2365     </level>
2366     <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sscc" >
2367     <option optionKey="12" pattern="urn:epc:id:sscc:([0-9]{12})\.[0-9]{5}"
2368 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2369     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2370 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2371     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2372 length="5" padChar="0" padDir="LEFT" name="serialref"/>
2373     </option>
2374     <option optionKey="11" pattern="urn:epc:id:sscc:([0-9]{11})\.[0-9]{6}"
2375 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2376     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2377 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2378     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2379 length="6" padChar="0" padDir="LEFT" name="serialref"/>
2380     </option>
2381     <option optionKey="10" pattern="urn:epc:id:sscc:([0-9]{10})\.[0-9]{7}"
2382 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2383     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2384 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2385     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2386 length="7" padChar="0" padDir="LEFT" name="serialref"/>
2387     </option>
2388     <option optionKey="9" pattern="urn:epc:id:sscc:([0-9]{9})\.[0-9]{8}"
2389 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2390     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2391 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2392     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2393 length="8" padChar="0" padDir="LEFT" name="serialref"/>
2394     </option>

```

```

2395     <option optionKey="8" pattern="urn:epc:id:sscc:([0-9]{8})\.([0-9]{9})"
2396 grammar="'urn:epc:id:sscc:' companyprefix '.' serialref" >
2397     <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2398 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2399     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2400 length="9" padChar="0" padDir="LEFT" name="serialref"/>
2401 </option>
2402     <option optionKey="7" pattern="urn:epc:id:sscc:([0-9]{7})\.([0-9]{10})"
2403 grammar="'urn:epc:id:sscc:' companyprefix '.' serialref" >
2404     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2405 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2406     <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2407 length="10" padChar="0" padDir="LEFT" name="serialref"/>
2408 </option>
2409     <option optionKey="6" pattern="urn:epc:id:sscc:([0-9]{6})\.([0-9]{11})"
2410 grammar="'urn:epc:id:sscc:' companyprefix '.' serialref" >
2411     <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2412 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2413     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2414 length="11" padChar="0" padDir="LEFT" name="serialref"/>
2415 </option>
2416 </level>
2417 <level type="LEGACY" prefixMatch="sscc"
2418 requiredParsingParameters="companyprefixlength" >
2419     <option optionKey="12" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2420 companyprefix serialrefremainder checkdigit" >
2421     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2422 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2423 </option>
2424     <option optionKey="11" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2425 companyprefix serialrefremainder checkdigit" >
2426     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2427 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2428 </option>
2429     <option optionKey="10" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2430 companyprefix serialrefremainder checkdigit" >
2431     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2432 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2433 </option>
2434     <option optionKey="9" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2435 companyprefix serialrefremainder checkdigit" >
2436     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2437 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2438 </option>
2439     <option optionKey="8" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2440 companyprefix serialrefremainder checkdigit" >
2441     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2442 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2443 </option>
2444     <option optionKey="7" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2445 companyprefix serialrefremainder checkdigit" >
2446     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2447 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2448 </option>
2449     <option optionKey="6" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2450 companyprefix serialrefremainder checkdigit" >
2451     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2452 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2453 </option>
2454     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="ssccprefixremainder"
2455 characterSet="[0-9]*" length="16" function="SUBSTR(sscc,1,16)"/>
2456     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="extensiondigit"
2457 characterSet="[0-9]*" length="1" function="SUBSTR(sscc,0,1)"/>
2458     <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="serialrefremainder"
2459 characterSet="[0-9]*" function="SUBSTR(ssccprefixremainder,companyprefixlength)"/>
2460     <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="serialref"
2461 characterSet="[0-9]*" function="CONCAT(extensiondigit,serialrefremainder)"/>
2462     <rule type="EXTRACT" inputFormat="STRING" seq="5" newFieldName="companyprefix"
2463 characterSet="[0-9]*" function="SUBSTR(ssccprefixremainder,0,companyprefixlength)"/>
2464     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="extensiondigit"
2465 characterSet="[0-9]*" length="1" function="SUBSTR(serialref,0,1)"/>

```

```

2466     <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="serialrefremainder"
2467 characterSet="[0-9]*" function="SUBSTR(serialref,1)"/>
2468     <rule type="FORMAT" inputFormat="STRING" seq="3" newFieldName="ssccprefix"
2469 characterSet="[0-9]*" length="17"
2470 function="CONCAT(extensiondigit,companyprefix,serialrefremainder)"/>
2471     <rule type="FORMAT" inputFormat="STRING" seq="4" newFieldName="checkdigit"
2472 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(ssccprefix)"/>
2473 </level>
2474 </scheme>
2475 </epcTagDataTranslation>

```

## 2476 8.5. SSCC-96 Serial Shipping Container Code for 96-bit EPCs

```

2477 <?xml version="1.0" encoding="UTF-8"?>
2478 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
2479 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
2480 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
2481   <scheme name="SSCC-96" optionKey="companyprefixlength" tagLength="96">
2482     <level type="BINARY" prefixMatch="00110001"
2483 requiredFormattingParameters="filter,taglength" >
2484       <option optionKey="12"
2485 pattern="00110001([01]{3})000([01]{40})([01]{18})000000000000000000000000"
2486 grammar="'00110001' filter '000' companyprefix serialref '000000000000000000000000'" >
2487         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2488 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2489         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
2490 bitLength="40" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2491         <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
2492 bitLength="18" length="5" padChar="0" padDir="LEFT" name="serialref"/>
2493       </option>
2494       <option optionKey="11"
2495 pattern="00110001([01]{3})001([01]{37})([01]{21})000000000000000000000000"
2496 grammar="'00110001' filter '001' companyprefix serialref '000000000000000000000000'" >
2497         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2498 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2499         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
2500 bitLength="37" length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2501         <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
2502 bitLength="21" length="6" padChar="0" padDir="LEFT" name="serialref"/>
2503       </option>
2504       <option optionKey="10"
2505 pattern="00110001([01]{3})010([01]{34})([01]{24})000000000000000000000000"
2506 grammar="'00110001' filter '010' companyprefix serialref '000000000000000000000000'" >
2507         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2508 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2509         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
2510 bitLength="34" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2511         <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
2512 bitLength="24" length="7" padChar="0" padDir="LEFT" name="serialref"/>
2513       </option>
2514       <option optionKey="9"
2515 pattern="00110001([01]{3})011([01]{30})([01]{28})000000000000000000000000"
2516 grammar="'00110001' filter '011' companyprefix serialref '000000000000000000000000'" >
2517         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2518 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2519         <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
2520 bitLength="30" length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2521         <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
2522 bitLength="28" length="8" padChar="0" padDir="LEFT" name="serialref"/>
2523       </option>
2524       <option optionKey="8"
2525 pattern="00110001([01]{3})100([01]{27})([01]{31})000000000000000000000000"
2526 grammar="'00110001' filter '100' companyprefix serialref '000000000000000000000000'" >
2527         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2528 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2529         <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
2530 bitLength="27" length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2531         <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
2532 bitLength="31" length="9" padChar="0" padDir="LEFT" name="serialref"/>
2533       </option>

```

```

2534     </option>
2535     <option optionKey="7"
2536 pattern="00110001([01]{3})101([01]{24})([01]{34})0000000000000000000000"
2537 grammar="'00110001' filter '101' companyprefix serialref '0000000000000000000000' >
2538     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2539 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2540     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
2541 bitLength="24" length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2542     <field seq="3" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
2543 bitLength="34" length="10" padChar="0" padDir="LEFT" name="serialref"/>
2544     </option>
2545     <option optionKey="6"
2546 pattern="00110001([01]{3})110([01]{20})([01]{38})0000000000000000000000"
2547 grammar="'00110001' filter '110' companyprefix serialref '0000000000000000000000' >
2548     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2549 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2550     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
2551 bitLength="20" length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2552     <field seq="3" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
2553 bitLength="38" length="11" padChar="0" padDir="LEFT" name="serialref"/>
2554     </option>
2555 </level>
2556 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sscc-96"
2557 requiredFormattingParameters="filter,taglength" >
2558     <option optionKey="12" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){12}]\.[(0-
2559 9]{5})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2560     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2561 length="1" padChar="0" padDir="LEFT" name="filter"/>
2562     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2563 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2564     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2565 length="5" padChar="0" padDir="LEFT" name="serialref"/>
2566     </option>
2567     <option optionKey="11" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){11}]\.[(0-
2568 9]{6})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2569     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2570 length="1" padChar="0" padDir="LEFT" name="filter"/>
2571     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2572 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2573     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2574 length="6" padChar="0" padDir="LEFT" name="serialref"/>
2575     </option>
2576     <option optionKey="10" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){10}]\.[(0-
2577 9]{7})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2578     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2579 length="1" padChar="0" padDir="LEFT" name="filter"/>
2580     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2581 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2582     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2583 length="7" padChar="0" padDir="LEFT" name="serialref"/>
2584     </option>
2585     <option optionKey="9" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){9}]\.[(0-
2586 9){8})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2587     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2588 length="1" padChar="0" padDir="LEFT" name="filter"/>
2589     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2590 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2591     <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2592 length="8" padChar="0" padDir="LEFT" name="serialref"/>
2593     </option>
2594     <option optionKey="8" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){8}]\.[(0-
2595 9){9})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2596     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2597 length="1" padChar="0" padDir="LEFT" name="filter"/>
2598     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2599 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2600     <field seq="3" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2601 length="9" padChar="0" padDir="LEFT" name="serialref"/>
2602     </option>
2603     <option optionKey="7" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.[(0-9){7}]\.[(0-
2604 9){10})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >

```

```

2605     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2606 length="1" padChar="0" padDir="LEFT" name="filter"/>
2607     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2608 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2609     <field seq="3" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2610 length="10" padChar="0" padDir="LEFT" name="serialref"/>
2611 </option>
2612 <option optionKey="6" pattern="urn:epc:tag:sscc-96:([0-7]{1})\.([0-9]{6})\.([0-
2613 9]{11})" grammar="urn:epc:tag:sscc-96:' filter '.' companyprefix '.' serialref" >
2614     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2615 length="1" padChar="0" padDir="LEFT" name="filter"/>
2616     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2617 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2618     <field seq="3" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2619 length="11" padChar="0" padDir="LEFT" name="serialref"/>
2620 </option>
2621 </level>
2622 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sscc" >
2623     <option optionKey="12" pattern="urn:epc:id:sscc:([0-9]{12})\.([0-9]{5})"
2624 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2625     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2626 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2627     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2628 length="5" padChar="0" padDir="LEFT" name="serialref"/>
2629 </option>
2630     <option optionKey="11" pattern="urn:epc:id:sscc:([0-9]{11})\.([0-9]{6})"
2631 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2632     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2633 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2634     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2635 length="6" padChar="0" padDir="LEFT" name="serialref"/>
2636 </option>
2637     <option optionKey="10" pattern="urn:epc:id:sscc:([0-9]{10})\.([0-9]{7})"
2638 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2639     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2640 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2641     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2642 length="7" padChar="0" padDir="LEFT" name="serialref"/>
2643 </option>
2644     <option optionKey="9" pattern="urn:epc:id:sscc:([0-9]{9})\.([0-9]{8})"
2645 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2646     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2647 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2648     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2649 length="8" padChar="0" padDir="LEFT" name="serialref"/>
2650 </option>
2651     <option optionKey="8" pattern="urn:epc:id:sscc:([0-9]{8})\.([0-9]{9})"
2652 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2653     <field seq="1" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
2654 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2655     <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2656 length="9" padChar="0" padDir="LEFT" name="serialref"/>
2657 </option>
2658     <option optionKey="7" pattern="urn:epc:id:sscc:([0-9]{7})\.([0-9]{10})"
2659 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2660     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2661 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2662     <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
2663 length="10" padChar="0" padDir="LEFT" name="serialref"/>
2664 </option>
2665     <option optionKey="6" pattern="urn:epc:id:sscc:([0-9]{6})\.([0-9]{11})"
2666 grammar="urn:epc:id:sscc:' companyprefix '.' serialref" >
2667     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2668 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2669     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2670 length="11" padChar="0" padDir="LEFT" name="serialref"/>
2671 </option>
2672 </level>
2673 <level type="LEGACY" prefixMatch="sscc="
2674 requiredParsingParameters="companyprefixlength" >

```



```

2675     <option optionKey="12" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2676 companyprefix serialrefremainder checkdigit" >
2677     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2678 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2679     </option>
2680     <option optionKey="11" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2681 companyprefix serialrefremainder checkdigit" >
2682     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2683 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2684     </option>
2685     <option optionKey="10" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2686 companyprefix serialrefremainder checkdigit" >
2687     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2688 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2689     </option>
2690     <option optionKey="9" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2691 companyprefix serialrefremainder checkdigit" >
2692     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2693 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2694     </option>
2695     <option optionKey="8" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2696 companyprefix serialrefremainder checkdigit" >
2697     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2698 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2699     </option>
2700     <option optionKey="7" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2701 companyprefix serialrefremainder checkdigit" >
2702     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2703 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2704     </option>
2705     <option optionKey="6" pattern="sscc=([0-9]{18})" grammar="'sscc=' extensiondigit
2706 companyprefix serialrefremainder checkdigit" >
2707     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999999999"
2708 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="sscc"/>
2709     </option>
2710     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="ssccprefixremainder"
2711 characterSet="[0-9]*" length="16" function="SUBSTR(sscc,1,16)"/>
2712     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="extensiondigit"
2713 characterSet="[0-9]*" length="1" function="SUBSTR(sscc,0,1)"/>
2714     <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="serialrefremainder"
2715 characterSet="[0-9]*" function="SUBSTR(ssccprefixremainder,companyprefixlength)"/>
2716     <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="serialref"
2717 characterSet="[0-9]*" function="CONCAT(extensiondigit,serialrefremainder)"/>
2718     <rule type="EXTRACT" inputFormat="STRING" seq="5" newFieldName="companyprefix"
2719 characterSet="[0-9]*" function="SUBSTR(ssccprefixremainder,0,companyprefixlength)"/>
2720     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="extensiondigit"
2721 characterSet="[0-9]*" length="1" function="SUBSTR(serialref,0,1)"/>
2722     <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="serialrefremainder"
2723 characterSet="[0-9]*" function="SUBSTR(serialref,1)"/>
2724     <rule type="FORMAT" inputFormat="STRING" seq="3" newFieldName="ssccprefix"
2725 characterSet="[0-9]*" length="17"
2726 function="CONCAT(extensiondigit,companyprefix,serialrefremainder)"/>
2727     <rule type="FORMAT" inputFormat="STRING" seq="4" newFieldName="checkdigit"
2728 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(ssccprefix)"/>
2729     </level>
2730 </scheme>
2731 </epcTagDataTranslation>
2732

```

## 2733 8.6. SGLN-64 Serialized Global Location Number for 64-bit 2734 EPCs

```

2735 <?xml version="1.0" encoding="UTF-8"?>
2736 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
2737 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
2738 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
2739   <scheme name="SGLN-64" optionKey="companyprefixlength" tagLength="64" >
2740     <level type="BINARY" prefixMatch="00001001"
2741     requiredFormattingParameters="filter,taglength" >

```

```

2742     <option optionKey="12" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2743 grammar="'00001001' filter companyprefixindex locationref serial" >
2744     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2745 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2746     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2747 bitLength="14" name="companyprefixindex"/>
2748     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2749 bitLength="20" length="0" padChar="0" padDir="LEFT" name="locationref"/>
2750     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2751 bitLength="19" name="serial"/>
2752 </option>
2753     <option optionKey="11" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2754 grammar="'00001001' filter companyprefixindex locationref serial" >
2755     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2756 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2757     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2758 bitLength="14" name="companyprefixindex"/>
2759     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2760 bitLength="20" length="1" padChar="0" padDir="LEFT" name="locationref"/>
2761     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2762 bitLength="19" name="serial"/>
2763 </option>
2764     <option optionKey="10" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2765 grammar="'00001001' filter companyprefixindex locationref serial" >
2766     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2767 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2768     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2769 bitLength="14" name="companyprefixindex"/>
2770     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2771 bitLength="20" length="2" padChar="0" padDir="LEFT" name="locationref"/>
2772     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2773 bitLength="19" name="serial"/>
2774 </option>
2775     <option optionKey="9" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2776 grammar="'00001001' filter companyprefixindex locationref serial" >
2777     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2778 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2779     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2780 bitLength="14" name="companyprefixindex"/>
2781     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2782 bitLength="20" length="3" padChar="0" padDir="LEFT" name="locationref"/>
2783     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2784 bitLength="19" name="serial"/>
2785 </option>
2786     <option optionKey="8" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2787 grammar="'00001001' filter companyprefixindex locationref serial" >
2788     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2789 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2790     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2791 bitLength="14" name="companyprefixindex"/>
2792     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2793 bitLength="20" length="4" padChar="0" padDir="LEFT" name="locationref"/>
2794     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2795 bitLength="19" name="serial"/>
2796 </option>
2797     <option optionKey="7" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2798 grammar="'00001001' filter companyprefixindex locationref serial" >
2799     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2800 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
2801     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2802 bitLength="14" name="companyprefixindex"/>
2803     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2804 bitLength="20" length="5" padChar="0" padDir="LEFT" name="locationref"/>
2805     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2806 bitLength="19" name="serial"/>
2807 </option>
2808     <option optionKey="6" pattern="00001001([01]{3})([01]{14})([01]{20})([01]{19})"
2809 grammar="'00001001' filter companyprefixindex locationref serial" >
2810     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
2811 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>

```

```

2812     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
2813 bitLength="14" name="companyprefixindex"/>
2814     <field seq="3" decimalMinimum="0" decimalMaximum="1048575" characterSet="[01]*"
2815 bitLength="20" length="6" padChar="0" padDir="LEFT" name="locationref"/>
2816     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
2817 bitLength="19" name="serial"/>
2818 </option>
2819     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="companyprefix"
2820 characterSet="[0-9]*"
2821 function="TABLELOOKUP(companyprefixindex,tdt64bitcpi,companyprefixindex,companyprefix)"
2822 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
2823 tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"
2824 tableParams="companyprefixindex"/>
2825     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefixlength"
2826 characterSet="[0-9]*" function="LENGTH(companyprefix)"/>
2827     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="companyprefixindex"
2828 characterSet="[0-9]*"
2829 function="TABLELOOKUP(companyprefix,tdt64bitcpi,companyprefix,companyprefixindex)"
2830 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
2831 tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index" tableParams="companyprefix"/>
2832 </level>
2833 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sgln-64"
2834 requiredFormattingParameters="filter,taglength" >
2835     <option optionKey="12" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9){12}]\.[(0-
2836 9]{0})\.[(0-9)*]" grammar="urn:epc:tag:sgln-64: filter '.' companyprefix '.'
2837 locationref '.' serial" >
2838         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2839 length="1" padChar="0" padDir="LEFT" name="filter"/>
2840         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-
2841 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2842         <field seq="3" decimalMinimum="0" decimalMaximum="1" characterSet="[0-9]*"
2843 length="0" padChar="0" padDir="LEFT" name="locationref"/>
2844         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2845 name="serial"/>
2846     </option>
2847     <option optionKey="11" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9){11}]\.[(0-
2848 9]{1})\.[(0-9)*]" grammar="urn:epc:tag:sgln-64: filter '.' companyprefix '.'
2849 locationref '.' serial" >
2850         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2851 length="1" padChar="0" padDir="LEFT" name="filter"/>
2852         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2853 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2854         <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
2855 length="1" padChar="0" padDir="LEFT" name="locationref"/>
2856         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2857 name="serial"/>
2858     </option>
2859     <option optionKey="10" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9){10}]\.[(0-
2860 9){2}]\.[(0-9)*]" grammar="urn:epc:tag:sgln-64: filter '.' companyprefix '.'
2861 locationref '.' serial" >
2862         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2863 length="1" padChar="0" padDir="LEFT" name="filter"/>
2864         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2865 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2866         <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
2867 length="2" padChar="0" padDir="LEFT" name="locationref"/>
2868         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2869 name="serial"/>
2870     </option>
2871     <option optionKey="9" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9){9}]\.[(0-
2872 9){3}]\.[(0-9)*]" grammar="urn:epc:tag:sgln-64: filter '.' companyprefix '.'
2873 locationref '.' serial" >
2874         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2875 length="1" padChar="0" padDir="LEFT" name="filter"/>
2876         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2877 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2878         <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
2879 length="3" padChar="0" padDir="LEFT" name="locationref"/>
2880         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2881 name="serial"/>
2882     </option>

```

```

2883     <option optionKey="8" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9]{8})\.[(0-
2884 9]{4})\.[(0-9]*)" grammar="urn:epc:tag:sgln-64:' filter '.' companyprefix '.'
2885 locationref '.' serial" >
2886     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2887 length="1" padChar="0" padDir="LEFT" name="filter"/>
2888     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2889 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2890     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
2891 length="4" padChar="0" padDir="LEFT" name="locationref"/>
2892     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2893 name="serial"/>
2894 </option>
2895     <option optionKey="7" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9]{7})\.[(0-
2896 9]{5})\.[(0-9]*)" grammar="urn:epc:tag:sgln-64:' filter '.' companyprefix '.'
2897 locationref '.' serial" >
2898     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2899 length="1" padChar="0" padDir="LEFT" name="filter"/>
2900     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2901 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2902     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2903 length="5" padChar="0" padDir="LEFT" name="locationref"/>
2904     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2905 name="serial"/>
2906 </option>
2907     <option optionKey="6" pattern="urn:epc:tag:sgln-64:([0-7]{1})\.[(0-9]{6})\.[(0-
2908 9]{6})\.[(0-9]*)" grammar="urn:epc:tag:sgln-64:' filter '.' companyprefix '.'
2909 locationref '.' serial" >
2910     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
2911 length="1" padChar="0" padDir="LEFT" name="filter"/>
2912     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2913 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2914     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2915 length="6" padChar="0" padDir="LEFT" name="locationref"/>
2916     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2917 name="serial"/>
2918 </option>
2919 </level>
2920 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sgln" >
2921     <option optionKey="12" pattern="urn:epc:id:sgln:([0-9]{12})\.[(0-9]{0})\.[(0-9]*)"
2922 grammar="urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2923     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
2924 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
2925     <field seq="2" decimalMinimum="0" decimalMaximum="1" characterSet="[0-9]*"
2926 length="0" padChar="0" padDir="LEFT" name="locationref"/>
2927     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2928 name="serial"/>
2929 </option>
2930     <option optionKey="11" pattern="urn:epc:id:sgln:([0-9]{11})\.[(0-9]{1})\.[(0-9]*)"
2931 grammar="urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2932     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2933 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
2934     <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
2935 length="1" padChar="0" padDir="LEFT" name="locationref"/>
2936     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2937 name="serial"/>
2938 </option>
2939     <option optionKey="10" pattern="urn:epc:id:sgln:([0-9]{10})\.[(0-9]{2})\.[(0-9]*)"
2940 grammar="urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2941     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
2942 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
2943     <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
2944 length="2" padChar="0" padDir="LEFT" name="locationref"/>
2945     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2946 name="serial"/>
2947 </option>
2948     <option optionKey="9" pattern="urn:epc:id:sgln:([0-9]{9})\.[(0-9]{3})\.[(0-9]*)"
2949 grammar="urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2950     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
2951 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
2952     <field seq="2" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
2953 length="3" padChar="0" padDir="LEFT" name="locationref"/>

```

```

2954     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2955     name="serial"/>
2956   </option>
2957   <option optionKey="8" pattern="urn:epc:id:sgln:([0-9]{8})\.([0-9]{4})\.([0-9]*)"
2958   grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2959     <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
2960     length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
2961     <field seq="2" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
2962     length="4" padChar="0" padDir="LEFT" name="locationref"/>
2963     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2964     name="serial"/>
2965   </option>
2966   <option optionKey="7" pattern="urn:epc:id:sgln:([0-9]{7})\.([0-9]{5})\.([0-9]*)"
2967   grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2968     <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
2969     length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
2970     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
2971     length="5" padChar="0" padDir="LEFT" name="locationref"/>
2972     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2973     name="serial"/>
2974   </option>
2975   <option optionKey="6" pattern="urn:epc:id:sgln:([0-9]{6})\.([0-9]{6})\.([0-9]*)"
2976   grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
2977     <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2978     length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
2979     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
2980     length="6" padChar="0" padDir="LEFT" name="locationref"/>
2981     <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2982     name="serial"/>
2983   </option>
2984 </level>
2985 <level type="LEGACY" prefixMatch="gln=" requiredParsingParameters="companyprefixlength"
2986 >
2987   <option optionKey="12" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
2988   companyprefix locationref checkdigit ';serial=' serial" >
2989     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
2990 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
2991     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2992     name="serial"/>
2993   </option>
2994   <option optionKey="11" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
2995   companyprefix locationref checkdigit ';serial=' serial" >
2996     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
2997 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
2998     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
2999     name="serial"/>
3000   </option>
3001   <option optionKey="10" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3002   companyprefix locationref checkdigit ';serial=' serial" >
3003     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
3004 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3005     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3006     name="serial"/>
3007   </option>
3008   <option optionKey="9" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3009   companyprefix locationref checkdigit ';serial=' serial" >
3010     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
3011 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3012     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3013     name="serial"/>
3014   </option>
3015   <option optionKey="8" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3016   companyprefix locationref checkdigit ';serial=' serial" >
3017     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999" characterSet="[0-
3018 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3019     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3020     name="serial"/>
3021   </option>
3022   <option optionKey="7" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3023   companyprefix locationref checkdigit ';serial=' serial" >

```

```

3024     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3025 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3026     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3027 name="serial"/>
3028   </option>
3029   <option optionKey="6" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3030 companyprefix locationref checkdigit 'serial' ">
3031     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3032 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3033     <field seq="2" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3034 name="serial"/>
3035   </option>
3036   <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="glnprefixremainder"
3037 characterSet="[0-9]*" length="12" function="SUBSTR(gln,0,12)"/>
3038   <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="locationref"
3039 characterSet="[0-9]*" function="SUBSTR(glnprefixremainder,companyprefixlength)"/>
3040   <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="companyprefix"
3041 characterSet="[0-9]*" function="SUBSTR(glnprefixremainder,0,companyprefixlength)"/>
3042   <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="glnprefix"
3043 characterSet="[0-9]*" length="12" function="CONCAT(companyprefix,locationref)"/>
3044   <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="checkdigit"
3045 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(glnprefix)"/>
3046 </level>
3047 </scheme>
3048 </epcTagDataTranslation>

```

## 3049 8.7. SGLN-96 Serialized Global Location Number for 96-bit 3050 EPCs

```

3051 <?xml version="1.0" encoding="UTF-8"?>
3052 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
3053 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3054 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
3055   <scheme name="SGLN-96" optionKey="companyprefixlength" tagLength="96" >
3056     <level type="BINARY" prefixMatch="00110010"
3057 requiredFormattingParameters="filter,taglength" >
3058       <option optionKey="12" pattern="00110010([01]{3})000([01]{40})([01]{1})([01]{41})"
3059 grammar="'00110010' filter '000' companyprefix locationref serial" >
3060         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3061 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3062         <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[01]*"
3063 bitLength="40" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3064         <field seq="3" characterSet="[01]*" bitLength="1" length="0" padChar="0"
3065 padDir="LEFT" name="locationref"/>
3066         <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3067 characterSet="[01]*" bitLength="41" name="serial"/>
3068       </option>
3069       <option optionKey="11" pattern="00110010([01]{3})001([01]{37})([01]{4})([01]{41})"
3070 grammar="'00110010' filter '001' companyprefix locationref serial" >
3071         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3072 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3073         <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[01]*"
3074 bitLength="37" length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3075         <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[01]*"
3076 bitLength="4" length="1" padChar="0" padDir="LEFT" name="locationref"/>
3077         <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3078 characterSet="[01]*" bitLength="41" name="serial"/>
3079       </option>
3080       <option optionKey="10" pattern="00110010([01]{3})010([01]{34})([01]{7})([01]{41})"
3081 grammar="'00110010' filter '010' companyprefix locationref serial" >
3082         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3083 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3084         <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[01]*"
3085 bitLength="34" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3086         <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[01]*"
3087 bitLength="7" length="2" padChar="0" padDir="LEFT" name="locationref"/>
3088         <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3089 characterSet="[01]*" bitLength="41" name="serial"/>
3090       </option>

```

```

3091     <option optionKey="9" pattern="00110010([01]{3})011([01]{30})([01]{11})([01]{41})"
3092 grammar="'00110010' filter '011' companyprefix locationref serial" >
3093     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3094 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3095     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
3096 bitLength="30" length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3097     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[01]*"
3098 bitLength="11" length="3" padChar="0" padDir="LEFT" name="locationref"/>
3099     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3100 characterSet="[01]*" bitLength="41" name="serial"/>
3101 </option>
3102     <option optionKey="8" pattern="00110010([01]{3})100([01]{27})([01]{14})([01]{41})"
3103 grammar="'00110010' filter '100' companyprefix locationref serial" >
3104     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3105 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3106     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[01]*"
3107 bitLength="27" length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3108     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[01]*"
3109 bitLength="14" length="4" padChar="0" padDir="LEFT" name="locationref"/>
3110     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3111 characterSet="[01]*" bitLength="41" name="serial"/>
3112 </option>
3113     <option optionKey="7" pattern="00110010([01]{3})101([01]{24})([01]{17})([01]{41})"
3114 grammar="'00110010' filter '101' companyprefix locationref serial" >
3115     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3116 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3117     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
3118 bitLength="24" length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3119     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
3120 bitLength="17" length="5" padChar="0" padDir="LEFT" name="locationref"/>
3121     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3122 characterSet="[01]*" bitLength="41" name="serial"/>
3123 </option>
3124     <option optionKey="6" pattern="00110010([01]{3})110([01]{20})([01]{21})([01]{41})"
3125 grammar="'00110010' filter '110' companyprefix locationref serial" >
3126     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3127 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3128     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
3129 bitLength="20" length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3130     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
3131 bitLength="21" length="6" padChar="0" padDir="LEFT" name="locationref"/>
3132     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551"
3133 characterSet="[01]*" bitLength="41" name="serial"/>
3134 </option>
3135 </level>
3136 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:sgln-96"
3137 requiredFormattingParameters="filter,taglength" >
3138     <option optionKey="12" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.\([0-9]{12})\.\([0-
3139 9]{0})\.\([0-9]*" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3140 locationref '.' serial" >
3141     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3142 length="1" padChar="0" padDir="LEFT" name="filter"/>
3143     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3144 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3145     <field seq="3" decimalMinimum="0" decimalMaximum="1" characterSet="[0-9]*"
3146 length="0" padChar="0" padDir="LEFT" name="locationref"/>
3147     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3148 9]*" name="serial"/>
3149 </option>
3150     <option optionKey="11" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.\([0-9]{11})\.\([0-
3151 9]{1})\.\([0-9]*" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3152 locationref '.' serial" >
3153     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3154 length="1" padChar="0" padDir="LEFT" name="filter"/>
3155     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3156 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3157     <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3158 length="1" padChar="0" padDir="LEFT" name="locationref"/>
3159     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3160 9]*" name="serial"/>
3161 </option>

```

```

3162     <option optionKey="10" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.[(0-9]{10})\.[(0-
3163 9]{2})\.[(0-9]*)" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3164 locationref '.' serial" >
3165     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3166 length="1" padChar="0" padDir="LEFT" name="filter"/>
3167     <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3168 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3169     <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3170 length="2" padChar="0" padDir="LEFT" name="locationref"/>
3171     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3172 9]*" name="serial"/>
3173     </option>
3174     <option optionKey="9" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.[(0-9]{9})\.[(0-
3175 9]{3})\.[(0-9]*)" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3176 locationref '.' serial" >
3177     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3178 length="1" padChar="0" padDir="LEFT" name="filter"/>
3179     <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3180 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3181     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
3182 length="3" padChar="0" padDir="LEFT" name="locationref"/>
3183     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3184 9]*" name="serial"/>
3185     </option>
3186     <option optionKey="8" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.[(0-9]{8})\.[(0-
3187 9]{4})\.[(0-9]*)" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3188 locationref '.' serial" >
3189     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3190 length="1" padChar="0" padDir="LEFT" name="filter"/>
3191     <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3192 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3193     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
3194 length="4" padChar="0" padDir="LEFT" name="locationref"/>
3195     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3196 9]*" name="serial"/>
3197     </option>
3198     <option optionKey="7" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.[(0-9]{7})\.[(0-
3199 9]{5})\.[(0-9]*)" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3200 locationref '.' serial" >
3201     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3202 length="1" padChar="0" padDir="LEFT" name="filter"/>
3203     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
3204 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3205     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
3206 length="5" padChar="0" padDir="LEFT" name="locationref"/>
3207     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3208 9]*" name="serial"/>
3209     </option>
3210     <option optionKey="6" pattern="urn:epc:tag:sgln-96:([0-7]{1})\.[(0-9]{6})\.[(0-
3211 9]{6})\.[(0-9]*)" grammar="urn:epc:tag:sgln-96:' filter '.' companyprefix '.'
3212 locationref '.' serial" >
3213     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3214 length="1" padChar="0" padDir="LEFT" name="filter"/>
3215     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
3216 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3217     <field seq="3" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
3218 length="6" padChar="0" padDir="LEFT" name="locationref"/>
3219     <field seq="4" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3220 9]*" name="serial"/>
3221     </option>
3222     </level>
3223     <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:sgln" >
3224     <option optionKey="12" pattern="urn:epc:id:sgln:([0-9]{12})\.[(0-9]{0})\.[(0-9]*)"
3225 grammar="urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3226     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
3227 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3228     <field seq="2" decimalMinimum="0" decimalMaximum="1" characterSet="[0-9]*"
3229 length="0" padChar="0" padDir="LEFT" name="locationref"/>
3230     <field seq="3" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3231 9]*" name="serial"/>
3232     </option>

```



```

3233     <option optionKey="11" pattern="urn:epc:id:sgln:([0-9]{11})\.[0-9]{1}\.[0-9]*"
3234 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3235     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3236 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3237     <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3238 length="1" padChar="0" padDir="LEFT" name="locationref"/>
3239     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3240 9]*" name="serial"/>
3241     </option>
3242     <option optionKey="10" pattern="urn:epc:id:sgln:([0-9]{10})\.[0-9]{2}\.[0-9]*"
3243 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3244     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3245 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3246     <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3247 length="2" padChar="0" padDir="LEFT" name="locationref"/>
3248     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3249 9]*" name="serial"/>
3250     </option>
3251     <option optionKey="9" pattern="urn:epc:id:sgln:([0-9]{9})\.[0-9]{3}\.[0-9]*"
3252 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3253     <field seq="1" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3254 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3255     <field seq="2" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
3256 length="3" padChar="0" padDir="LEFT" name="locationref"/>
3257     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3258 9]*" name="serial"/>
3259     </option>
3260     <option optionKey="8" pattern="urn:epc:id:sgln:([0-9]{8})\.[0-9]{4}\.[0-9]*"
3261 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3262     <field seq="1" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3263 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3264     <field seq="2" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
3265 length="4" padChar="0" padDir="LEFT" name="locationref"/>
3266     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3267 9]*" name="serial"/>
3268     </option>
3269     <option optionKey="7" pattern="urn:epc:id:sgln:([0-9]{7})\.[0-9]{5}\.[0-9]*"
3270 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3271     <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
3272 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3273     <field seq="2" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
3274 length="5" padChar="0" padDir="LEFT" name="locationref"/>
3275     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3276 9]*" name="serial"/>
3277     </option>
3278     <option optionKey="6" pattern="urn:epc:id:sgln:([0-9]{6})\.[0-9]{6}\.[0-9]*"
3279 grammar="'urn:epc:id:sgln:' companyprefix '.' locationref '.' serial" >
3280     <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
3281 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3282     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
3283 length="6" padChar="0" padDir="LEFT" name="locationref"/>
3284     <field seq="3" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3285 9]*" name="serial"/>
3286     </option>
3287     </level>
3288     <level type="LEGACY" prefixMatch="gln=" requiredParsingParameters="companyprefixlength"
3289     >
3290     <option optionKey="12" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln=' gln
3291     ';serial=' serial" >
3292     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
3293 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3294     <field seq="2" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3295 9]*" name="serial"/>
3296     </option>
3297     <option optionKey="11" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3298     companyprefix locationref checkdigit ';serial=' serial" >
3299     <field seq="1" decimalMinimum="0" decimalMaximum="99999999999999" characterSet="[0-
3300 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3301     <field seq="2" decimalMinimum="0" decimalMaximum="219902325551" characterSet="[0-
3302 9]*" name="serial"/>
3303     </option>

```

```

3304     <option optionKey="10" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3305 companyprefix locationref checkdigit ';serial=' serial" >
3306     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3307 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3308     <field seq="2" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3309 9]*" name="serial"/>
3310     </option>
3311     <option optionKey="9" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3312 companyprefix locationref checkdigit ';serial=' serial" >
3313     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3314 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3315     <field seq="2" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3316 9]*" name="serial"/>
3317     </option>
3318     <option optionKey="8" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3319 companyprefix locationref checkdigit ';serial=' serial" >
3320     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3321 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3322     <field seq="2" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3323 9]*" name="serial"/>
3324     </option>
3325     <option optionKey="7" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3326 companyprefix locationref checkdigit ';serial=' serial" >
3327     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3328 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3329     <field seq="2" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3330 9]*" name="serial"/>
3331     </option>
3332     <option optionKey="6" pattern="gln=([0-9]{13});serial=([0-9]*)" grammar="'gln='
3333 companyprefix locationref checkdigit ';serial=' serial" >
3334     <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3335 9]*" length="13" padChar="0" padDir="LEFT" name="gln"/>
3336     <field seq="2" decimalMinimum="0" decimalMaximum="2199023255551" characterSet="[0-
3337 9]*" name="serial"/>
3338     </option>
3339     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="glnprefixremainder"
3340 characterSet="[0-9]*" length="12" function="SUBSTR(gln,0,12)"/>
3341     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="locationref"
3342 characterSet="[0-9]*" function="SUBSTR(glnprefixremainder,companyprefixlength)"/>
3343     <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="companyprefix"
3344 characterSet="[0-9]*" function="SUBSTR(glnprefixremainder,0,companyprefixlength)"/>
3345     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="glnprefix"
3346 characterSet="[0-9]*" length="12" function="CONCAT(companyprefix,locationref)"/>
3347     <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="checkdigit"
3348 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(glnprefix)"/>
3349     </level>
3350 </scheme>
3351 </epcTagDataTranslation>

```

## 3352 8.8. GRAI-64 Global Returnable Asset Identifier for 64-bit 3353 EPCs

```

3354 <?xml version="1.0" encoding="UTF-8"?>
3355 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
3356 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3357 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
3358   <scheme name="GRAI-64" optionKey="companyprefixlength" tagLength="64" >
3359     <level type="BINARY" prefixMatch="00001010"
3360 requiredFormattingParameters="filter,taglength" >
3361       <option optionKey="12" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3362 grammar="'00001010' filter companyprefixindex assettype serial" >
3363         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3364 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3365         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3366 bitLength="14" name="companyprefixindex"/>
3367         <field seq="3" characterSet="[01]*" bitLength="20" length="0" padChar="0"
3368 padDir="LEFT" name="assettype"/>
3369         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3370 bitLength="19" name="serial"/>

```

```

3371     </option>
3372     <option optionKey="11" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3373     grammar="'00001010' filter companyprefixindex assettype serial" >
3374         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3375         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3376         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3377         bitLength="14" name="companyprefixindex"/>
3378         <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[01]*"
3379         bitLength="20" length="1" padChar="0" padDir="LEFT" name="assettype"/>
3380         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3381         bitLength="19" name="serial"/>
3382     </option>
3383     <option optionKey="10" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3384     grammar="'00001010' filter companyprefixindex assettype serial" >
3385         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3386         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3387         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3388         bitLength="14" name="companyprefixindex"/>
3389         <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[01]*"
3390         bitLength="20" length="2" padChar="0" padDir="LEFT" name="assettype"/>
3391         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3392         bitLength="19" name="serial"/>
3393     </option>
3394     <option optionKey="9" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3395     grammar="'00001010' filter companyprefixindex assettype serial" >
3396         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3397         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3398         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3399         bitLength="14" name="companyprefixindex"/>
3400         <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[01]*"
3401         bitLength="20" length="3" padChar="0" padDir="LEFT" name="assettype"/>
3402         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3403         bitLength="19" name="serial"/>
3404     </option>
3405     <option optionKey="8" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3406     grammar="'00001010' filter companyprefixindex assettype serial" >
3407         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3408         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3409         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3410         bitLength="14" name="companyprefixindex"/>
3411         <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[01]*"
3412         bitLength="20" length="4" padChar="0" padDir="LEFT" name="assettype"/>
3413         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3414         bitLength="19" name="serial"/>
3415     </option>
3416     <option optionKey="7" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3417     grammar="'00001010' filter companyprefixindex assettype serial" >
3418         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3419         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3420         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3421         bitLength="14" name="companyprefixindex"/>
3422         <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
3423         bitLength="20" length="5" padChar="0" padDir="LEFT" name="assettype"/>
3424         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3425         bitLength="19" name="serial"/>
3426     </option>
3427     <option optionKey="6" pattern="00001010([01]{3})([01]{14})([01]{20})([01]{19})"
3428     grammar="'00001010' filter companyprefixindex assettype serial" >
3429         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3430         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3431         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3432         bitLength="14" name="companyprefixindex"/>
3433         <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
3434         bitLength="20" length="6" padChar="0" padDir="LEFT" name="assettype"/>
3435         <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[01]*"
3436         bitLength="19" name="serial"/>
3437     </option>
3438     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="companyprefix"
3439     characterSet="[0-9]*"
3440     function="TABLELOOKUP(companyprefixindex,tdt64bitcpi,companyprefixindex,companyprefix)"
3441     tableURI="http://www.onsepc.com/ManagerTranslation.xml"

```

```

3442 tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"
3443 tableParams="companyprefixindex"/>
3444 <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefixlength"
3445 characterSet="[0-9]*" function="LENGTH(companyprefix)"/>
3446 <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="companyprefixindex"
3447 characterSet="[0-9]*"
3448 function="TABLELOOKUP(companyprefix,tdt64bitcpi,companyprefix,companyprefixindex)"
3449 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
3450 tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index" tableParams="companyprefix"/>
3451 </level>
3452 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:grai-64"
3453 requiredFormattingParameters="filter,taglength" >
3454 <option optionKey="12" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[(0-9){12}]\.[(0-
3455 9]{0})\.[(0-9)*]" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3456 '.' serial" >
3457 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3458 length="1" padChar="0" padDir="LEFT" name="filter"/>
3459 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-
3460 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3461 <field seq="3" characterSet="[0-9]*" length="0" padChar="0" padDir="LEFT"
3462 name="assettype"/>
3463 <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3464 name="serial"/>
3465 </option>
3466 <option optionKey="11" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[(0-9){11}]\.[(0-
3467 9]{1})\.[(0-9)*]" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3468 '.' serial" >
3469 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3470 length="1" padChar="0" padDir="LEFT" name="filter"/>
3471 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3472 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3473 <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3474 length="1" padChar="0" padDir="LEFT" name="assettype"/>
3475 <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3476 name="serial"/>
3477 </option>
3478 <option optionKey="10" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[(0-9){10}]\.[(0-
3479 9){2}]\.[(0-9)*]" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3480 '.' serial" >
3481 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3482 length="1" padChar="0" padDir="LEFT" name="filter"/>
3483 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3484 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3485 <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3486 length="2" padChar="0" padDir="LEFT" name="assettype"/>
3487 <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3488 name="serial"/>
3489 </option>
3490 <option optionKey="9" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[(0-9){9}]\.[(0-
3491 9){3}]\.[(0-9)*]" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3492 '.' serial" >
3493 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3494 length="1" padChar="0" padDir="LEFT" name="filter"/>
3495 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[0-9]*"
3496 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3497 <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
3498 length="3" padChar="0" padDir="LEFT" name="assettype"/>
3499 <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3500 name="serial"/>
3501 </option>
3502 <option optionKey="8" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[(0-9){8}]\.[(0-
3503 9){4}]\.[(0-9)*]" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3504 '.' serial" >
3505 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3506 length="1" padChar="0" padDir="LEFT" name="filter"/>
3507 <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
3508 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3509 <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
3510 length="4" padChar="0" padDir="LEFT" name="assettype"/>
3511 <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3512 name="serial"/>

```

```

3513     </option>
3514     <option optionKey="7" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[0-9]{7}\.[0-
3515 9]{5}\.[0-9]*" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3516 '.' serial" >
3517     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3518 length="1" padChar="0" padDir="LEFT" name="filter"/>
3519     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
3520 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3521     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
3522 length="5" padChar="0" padDir="LEFT" name="assettype"/>
3523     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3524 name="serial"/>
3525     </option>
3526     <option optionKey="6" pattern="urn:epc:tag:grai-64:([0-7]{1})\.[0-9]{6}\.[0-
3527 9]{6}\.[0-9]*" grammar="urn:epc:tag:grai-64:' filter '.' companyprefix '.' assettype
3528 '.' serial" >
3529     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3530 length="1" padChar="0" padDir="LEFT" name="filter"/>
3531     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
3532 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3533     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
3534 length="6" padChar="0" padDir="LEFT" name="assettype"/>
3535     <field seq="4" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3536 name="serial"/>
3537     </option>
3538 </level>
3539 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:grai" >
3540 <option optionKey="12" pattern="urn:epc:id:grai:([0-9]{12})\.[0-9]{0}\.[0-9]*"
3541 grammar="urn:epc:id:grai:' companyprefix '.' assettype '.' serial" >
3542 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3543 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3544 <field seq="2" characterSet="[0-9]*" length="0" padChar="0" padDir="LEFT"
3545 name="assettype"/>
3546 <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3547 name="serial"/>
3548 </option>
3549 <option optionKey="11" pattern="urn:epc:id:grai:([0-9]{11})\.[0-9]{1}\.[0-9]*"
3550 grammar="urn:epc:id:grai:' companyprefix '.' assettype '.' serial" >
3551 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3552 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3553 <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3554 length="1" padChar="0" padDir="LEFT" name="assettype"/>
3555 <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3556 name="serial"/>
3557 </option>
3558 <option optionKey="10" pattern="urn:epc:id:grai:([0-9]{10})\.[0-9]{2}\.[0-9]*"
3559 grammar="urn:epc:id:grai:' companyprefix '.' assettype '.' serial" >
3560 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3561 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3562 <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3563 length="2" padChar="0" padDir="LEFT" name="assettype"/>
3564 <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3565 name="serial"/>
3566 </option>
3567 <option optionKey="9" pattern="urn:epc:id:grai:([0-9]{9})\.[0-9]{3}\.[0-9]*"
3568 grammar="urn:epc:id:grai:' companyprefix '.' assettype '.' serial" >
3569 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3570 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3571 <field seq="2" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
3572 length="3" padChar="0" padDir="LEFT" name="assettype"/>
3573 <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3574 name="serial"/>
3575 </option>
3576 <option optionKey="8" pattern="urn:epc:id:grai:([0-9]{8})\.[0-9]{4}\.[0-9]*"
3577 grammar="urn:epc:id:grai:' companyprefix '.' assettype '.' serial" >
3578 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3579 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3580 <field seq="2" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
3581 length="4" padChar="0" padDir="LEFT" name="assettype"/>
3582 <field seq="3" decimalMinimum="0" decimalMaximum="524287" characterSet="[0-9]*"
3583 name="serial"/>

```



```

3654     <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="companyprefix"
3655 characterSet="[0-9]*" function="SUBSTR(graiprefixremainder,0,companyprefixlength)"/>
3656     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="graiprefix"
3657 characterSet="[0-9]*" length="13" function="CONCAT(0,companyprefix,assettype)"/>
3658     <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="checkdigit"
3659 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(graiprefix)"/>
3660   </level>
3661 </scheme>
3662 </epcTagDataTranslation>

```

## 3663 8.9. GRAI-96 Global Returnable Asset Identifier for 96-bit 3664 EPCs

```

3665 <?xml version="1.0" encoding="UTF-8"?>
3666 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
3667 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3668 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
3669   <scheme name="GRAI-96" optionKey="companyprefixlength" tagLength="96" >
3670     <level type="BINARY" prefixMatch="00110011"
3671 requiredFormattingParameters="filter,taglength" >
3672       <option optionKey="12" pattern="00110011([01]{3})000([01]{40})([01]{4})([01]{38})"
3673 grammar="'00110011' filter '000' companyprefix assettype serial" >
3674         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3675 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3676         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
3677 bitLength="40" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3678         <field seq="3" characterSet="[01]*" bitLength="4" length="0" padChar="0"
3679 padDir="LEFT" name="assettype"/>
3680         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3681 bitLength="38" name="serial"/>
3682       </option>
3683       <option optionKey="11" pattern="00110011([01]{3})001([01]{37})([01]{7})([01]{38})"
3684 grammar="'00110011' filter '001' companyprefix assettype serial" >
3685         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3686 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3687         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
3688 bitLength="37" length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3689         <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[01]*"
3690 bitLength="7" length="1" padChar="0" padDir="LEFT" name="assettype"/>
3691         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3692 bitLength="38" name="serial"/>
3693       </option>
3694       <option optionKey="10" pattern="00110011([01]{3})010([01]{34})([01]{10})([01]{38})"
3695 grammar="'00110011' filter '010' companyprefix assettype serial" >
3696         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3697 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3698         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
3699 bitLength="34" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3700         <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[01]*"
3701 bitLength="10" length="2" padChar="0" padDir="LEFT" name="assettype"/>
3702         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3703 bitLength="38" name="serial"/>
3704       </option>
3705       <option optionKey="9" pattern="00110011([01]{3})011([01]{30})([01]{14})([01]{38})"
3706 grammar="'00110011' filter '011' companyprefix assettype serial" >
3707         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3708 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3709         <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[01]*"
3710 bitLength="30" length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3711         <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[01]*"
3712 bitLength="14" length="3" padChar="0" padDir="LEFT" name="assettype"/>
3713         <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3714 bitLength="38" name="serial"/>
3715       </option>
3716       <option optionKey="8" pattern="00110011([01]{3})100([01]{27})([01]{17})([01]{38})"
3717 grammar="'00110011' filter '100' companyprefix assettype serial" >
3718         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3719 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>

```

```

3720     <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[01]*"
3721 bitLength="27" length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3722     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[01]*"
3723 bitLength="17" length="4" padChar="0" padDir="LEFT" name="assettype"/>
3724     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3725 bitLength="38" name="serial"/>
3726 </option>
3727 <option optionKey="7" pattern="00110011([01]{3})101([01]{24})([01]{20})([01]{38})"
3728 grammar="'00110011' filter '101' companyprefix assettype serial" >
3729     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3730 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3731     <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
3732 bitLength="24" length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3733     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[01]*"
3734 bitLength="20" length="5" padChar="0" padDir="LEFT" name="assettype"/>
3735     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3736 bitLength="38" name="serial"/>
3737 </option>
3738 <option optionKey="6" pattern="00110011([01]{3})110([01]{20})([01]{24})([01]{38})"
3739 grammar="'00110011' filter '110' companyprefix assettype serial" >
3740     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3741 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3742     <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
3743 bitLength="20" length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3744     <field seq="3" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
3745 bitLength="24" length="6" padChar="0" padDir="LEFT" name="assettype"/>
3746     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[01]*"
3747 bitLength="38" name="serial"/>
3748 </option>
3749 </level>
3750 <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:grai-96"
3751 requiredFormattingParameters="filter taglength" >
3752 <option optionKey="12" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[([0-9]{12})\.[([0-
3753 9]{0})\.[([0-9]*)" grammar="'urn:epc:tag:grai-96:' filter '.' companyprefix '.' assettype
3754 '.' serial" >
3755     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3756 length="1" padChar="0" padDir="LEFT" name="filter"/>
3757     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
3758 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3759     <field seq="3" characterSet="[0-9]*" length="0" padChar="0" padDir="LEFT"
3760 name="assettype"/>
3761     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3762 9]*" name="serial"/>
3763 </option>
3764 <option optionKey="11" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[([0-9]{11})\.[([0-
3765 9]{1})\.[([0-9]*)" grammar="'urn:epc:tag:grai-96:' filter '.' companyprefix '.' assettype
3766 '.' serial" >
3767     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3768 length="1" padChar="0" padDir="LEFT" name="filter"/>
3769     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3770 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3771     <field seq="3" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3772 length="1" padChar="0" padDir="LEFT" name="assettype"/>
3773     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3774 9]*" name="serial"/>
3775 </option>
3776 <option optionKey="10" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[([0-9]{10})\.[([0-
3777 9]{2})\.[([0-9]*)" grammar="'urn:epc:tag:grai-96:' filter '.' companyprefix '.' assettype
3778 '.' serial" >
3779     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3780 length="1" padChar="0" padDir="LEFT" name="filter"/>
3781     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
3782 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3783     <field seq="3" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3784 length="2" padChar="0" padDir="LEFT" name="assettype"/>
3785     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3786 9]*" name="serial"/>
3787 </option>
3788 <option optionKey="9" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[([0-9]{9})\.[([0-
3789 9]{3})\.[([0-9]*)" grammar="'urn:epc:tag:grai-96:' filter '.' companyprefix '.' assettype
3790 '.' serial" >

```



```

3791     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3792 length="1" padChar="0" padDir="LEFT" name="filter"/>
3793     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3794 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
3795     <field seq="3" decimalMinimum="0" decimalMaximum="999" characterSet="[0-9]*"
3796 length="3" padChar="0" padDir="LEFT" name="assettype"/>
3797     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3798 9]*" name="serial"/>
3799     </option>
3800     <option optionKey="8" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[0-9]{8}\.[0-
3801 9]{4}\.([0-9]*)" grammar="urn:epc:tag:grai-96: filter '.' companyprefix '.' assettype
3802 '.' serial" >
3803     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3804 length="1" padChar="0" padDir="LEFT" name="filter"/>
3805     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3806 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
3807     <field seq="3" decimalMinimum="0" decimalMaximum="9999" characterSet="[0-9]*"
3808 length="4" padChar="0" padDir="LEFT" name="assettype"/>
3809     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3810 9]*" name="serial"/>
3811     </option>
3812     <option optionKey="7" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[0-9]{7}\.[0-
3813 9]{5}\.([0-9]*)" grammar="urn:epc:tag:grai-96: filter '.' companyprefix '.' assettype
3814 '.' serial" >
3815     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3816 length="1" padChar="0" padDir="LEFT" name="filter"/>
3817     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3818 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
3819     <field seq="3" decimalMinimum="0" decimalMaximum="99999" characterSet="[0-9]*"
3820 length="5" padChar="0" padDir="LEFT" name="assettype"/>
3821     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3822 9]*" name="serial"/>
3823     </option>
3824     <option optionKey="6" pattern="urn:epc:tag:grai-96:([0-7]{1})\.[0-9]{6}\.[0-
3825 9]{6}\.([0-9]*)" grammar="urn:epc:tag:grai-96: filter '.' companyprefix '.' assettype
3826 '.' serial" >
3827     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
3828 length="1" padChar="0" padDir="LEFT" name="filter"/>
3829     <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3830 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
3831     <field seq="3" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
3832 length="6" padChar="0" padDir="LEFT" name="assettype"/>
3833     <field seq="4" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3834 9]*" name="serial"/>
3835     </option>
3836     </level>
3837     <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:grai" >
3838     <option optionKey="12" pattern="urn:epc:id:grai:([0-9]{12})\.[0-9]{0}\.[0-9]*"
3839 grammar="urn:epc:id:grai: companyprefix '.' assettype '.' serial" >
3840     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999" characterSet="[0-
3841 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
3842     <field seq="2" characterSet="[0-9]*" length="0" padChar="0" padDir="LEFT"
3843 name="assettype"/>
3844     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3845 9]*" name="serial"/>
3846     </option>
3847     <option optionKey="11" pattern="urn:epc:id:grai:([0-9]{11})\.[0-9]{1}\.[0-9]*"
3848 grammar="urn:epc:id:grai: companyprefix '.' assettype '.' serial" >
3849     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999" characterSet="[0-9]*"
3850 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
3851     <field seq="2" decimalMinimum="0" decimalMaximum="9" characterSet="[0-9]*"
3852 length="1" padChar="0" padDir="LEFT" name="assettype"/>
3853     <field seq="3" decimalMinimum="0" decimalMaximum="274877906943" characterSet="[0-
3854 9]*" name="serial"/>
3855     </option>
3856     <option optionKey="10" pattern="urn:epc:id:grai:([0-9]{10})\.[0-9]{2}\.[0-9]*"
3857 grammar="urn:epc:id:grai: companyprefix '.' assettype '.' serial" >
3858     <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999" characterSet="[0-9]*"
3859 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
3860     <field seq="2" decimalMinimum="0" decimalMaximum="99" characterSet="[0-9]*"
3861 length="2" padChar="0" padDir="LEFT" name="assettype"/>

```



```

3933      </option>
3934      <option optionKey="7" pattern="grai=([0-9]{15,30})" grammar="grai=' 0'
3935 companyprefix assettype checkdigit serial" >
3936      <field seq="1" decimalMinimum="0"
3937 decimalMaximum="99999999999999999999999999999999" characterSet="[0-9]*"
3938 name="grai"/>
3939      </option>
3940      <option optionKey="6" pattern="grai=([0-9]{15,30})" grammar="grai=' 0'
3941 companyprefix assettype checkdigit serial" >
3942      <field seq="1" decimalMinimum="0"
3943 decimalMaximum="99999999999999999999999999999999" characterSet="[0-9]*"
3944 name="grai"/>
3945      </option>
3946      <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="graiprefixremainder"
3947 characterSet="[0-9]*" length="12" function="SUBSTR(grai,1,12)"/>
3948      <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="assettype"
3949 characterSet="[0-9]*" function="SUBSTR(graiprefixremainder,companyprefixlength)"/>
3950      <rule type="EXTRACT" inputFormat="STRING" seq="3" newFieldName="serial"
3951 characterSet="[0-9]*" function="SUBSTR(grai,14)"/>
3952      <rule type="EXTRACT" inputFormat="STRING" seq="4" newFieldName="companyprefix"
3953 characterSet="[0-9]*" function="SUBSTR(graiprefixremainder,0,companyprefixlength)"/>
3954      <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="graiprefix"
3955 characterSet="[0-9]*" length="13" function="CONCAT(0,companyprefix,assettype)"/>
3956      <rule type="FORMAT" inputFormat="STRING" seq="2" newFieldName="checkdigit"
3957 characterSet="[0-9]*" length="1" function="GS1CHECKSUM(graiprefix)"/>
3958      </level>
3959      </scheme>
3960 </epcTagDataTranslation>

```

## 3961 **8.10. GIAI-64 Global Individual Asset Identifier for 64-bit EPCs**

```

3962 <?xml version="1.0" encoding="UTF-8"?>
3963 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
3964 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3965 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
3966   <scheme name="GIAI-64" optionKey="companyprefixlength" tagLength="64" >
3967     <level type="BINARY" prefixMatch="00001011"
3968     requiredFormattingParameters="filter,taglength" >
3969       <option optionKey="12" pattern="00001011{01}{3}{01}{14}{01}{39}"
3970 grammar="00001011 filter companyprefixindex indassetref" >
3971         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3972 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3973         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3974 bitLength="14" name="companyprefixindex"/>
3975         <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
3976 bitLength="39" length="12" padChar="0" padDir="LEFT" name="indassetref"/>
3977       </option>
3978       <option optionKey="11" pattern="00001011{01}{3}{01}{14}{01}{39}"
3979 grammar="00001011 filter companyprefixindex indassetref" >
3980         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3981 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3982         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3983 bitLength="14" name="companyprefixindex"/>
3984         <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
3985 bitLength="39" length="13" padChar="0" padDir="LEFT" name="indassetref"/>
3986       </option>
3987       <option optionKey="10" pattern="00001011{01}{3}{01}{14}{01}{39}"
3988 grammar="00001011 filter companyprefixindex indassetref" >
3989         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3990 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
3991         <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
3992 bitLength="14" name="companyprefixindex"/>
3993         <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
3994 bitLength="39" length="14" padChar="0" padDir="LEFT" name="indassetref"/>
3995       </option>
3996       <option optionKey="9" pattern="00001011{01}{3}{01}{14}{01}{39}"
3997 grammar="00001011 filter companyprefixindex indassetref" >
3998         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
3999 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>

```

```

4000     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
4001 bitLength="14" name="companyprefixindex"/>
4002     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
4003 bitLength="39" length="15" padChar="0" padDir="LEFT" name="indassetref"/>
4004     </option>
4005     <option optionKey="8" pattern="00001011([01]{3})([01]{14})([01]{39})"
4006 grammar="'00001011' filter companyprefixindex indassetref" >
4007     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4008 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4009     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
4010 bitLength="14" name="companyprefixindex"/>
4011     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
4012 bitLength="39" length="16" padChar="0" padDir="LEFT" name="indassetref"/>
4013     </option>
4014     <option optionKey="7" pattern="00001011([01]{3})([01]{14})([01]{39})"
4015 grammar="'00001011' filter companyprefixindex indassetref" >
4016     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4017 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4018     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
4019 bitLength="14" name="companyprefixindex"/>
4020     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
4021 bitLength="39" length="17" padChar="0" padDir="LEFT" name="indassetref"/>
4022     </option>
4023     <option optionKey="6" pattern="00001011([01]{3})([01]{14})([01]{39})"
4024 grammar="'00001011' filter companyprefixindex indassetref" >
4025     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4026 bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4027     <field seq="2" decimalMinimum="0" decimalMaximum="16383" characterSet="[01]*"
4028 bitLength="14" name="companyprefixindex"/>
4029     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[01]*"
4030 bitLength="39" length="18" padChar="0" padDir="LEFT" name="indassetref"/>
4031     </option>
4032     <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="companyprefix"
4033 characterSet="[0-9]*"
4034 function="TABLELOOKUP(companyprefixindex,tdt64bitcpi,companyprefixindex,companyprefix)"
4035 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
4036 tableXPath="/GEPC64Table/entry[@index='$1']/@companyPrefix"
4037 tableParams="companyprefixindex"/>
4038     <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefixlength"
4039 characterSet="[0-9]*" function="LENGTH(companyprefix)"/>
4040     <rule type="FORMAT" inputFormat="STRING" seq="1" newFieldName="companyprefixindex"
4041 characterSet="[0-9]*"
4042 function="TABLELOOKUP(companyprefix,tdt64bitcpi,companyprefix,companyprefixindex)"
4043 tableURI="http://www.onsepc.com/ManagerTranslation.xml"
4044 tableXPath="/GEPC64Table/entry[@companyPrefix='$1']/@index" tableParams="companyprefix"/>
4045     </level>
4046     <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:giai-64"
4047 requiredFormattingParameters="filter,taglength" >
4048     <option optionKey="12" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[(0-9){12}]\.[(0-
4049 9){12}]" grammar="'urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >
4050     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4051 length="1" padChar="0" padDir="LEFT" name="filter"/>
4052     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
4053 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
4054     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-
4055 9]*" length="12" padChar="0" padDir="LEFT" name="indassetref"/>
4056     </option>
4057     <option optionKey="11" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[(0-9){11}]\.[(0-
4058 9){13}]" grammar="'urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >
4059     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4060 length="1" padChar="0" padDir="LEFT" name="filter"/>
4061     <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
4062 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
4063     <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-
4064 9]*" length="13" padChar="0" padDir="LEFT" name="indassetref"/>
4065     </option>
4066     <option optionKey="10" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[(0-9){10}]\.[(0-
4067 9){14}]" grammar="'urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >
4068     <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4069 length="1" padChar="0" padDir="LEFT" name="filter"/>

```

4070 <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]\*"  
4071 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>  
4072 <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-  
4073 9]\*" length="14" padChar="0" padDir="LEFT" name="indassetref"/>  
4074 </option>  
4075 <option optionKey="9" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[0-9]{9}\.[0-  
4076 9]{15}" grammar="urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >  
4077 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]\*"  
4078 length="1" padChar="0" padDir="LEFT" name="filter"/>  
4079 <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]\*"  
4080 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>  
4081 <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-  
4082 9]\*" length="15" padChar="0" padDir="LEFT" name="indassetref"/>  
4083 </option>  
4084 <option optionKey="8" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[0-9]{8}\.[0-  
4085 9]{16}" grammar="urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >  
4086 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]\*"  
4087 length="1" padChar="0" padDir="LEFT" name="filter"/>  
4088 <field seq="2" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]\*"  
4089 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>  
4090 <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-  
4091 9]\*" length="16" padChar="0" padDir="LEFT" name="indassetref"/>  
4092 </option>  
4093 <option optionKey="7" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[0-9]{7}\.[0-  
4094 9]{17}" grammar="urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >  
4095 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]\*"  
4096 length="1" padChar="0" padDir="LEFT" name="filter"/>  
4097 <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]\*"  
4098 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>  
4099 <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-  
4100 9]\*" length="17" padChar="0" padDir="LEFT" name="indassetref"/>  
4101 </option>  
4102 <option optionKey="6" pattern="urn:epc:tag:giai-64:([0-7]{1})\.[0-9]{6}\.[0-  
4103 9]{18}" grammar="urn:epc:tag:giai-64:' filter '.' companyprefix '.' indassetref" >  
4104 <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]\*"  
4105 length="1" padChar="0" padDir="LEFT" name="filter"/>  
4106 <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]\*"  
4107 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>  
4108 <field seq="3" decimalMinimum="0" decimalMaximum="549755813887" characterSet="[0-  
4109 9]\*" length="18" padChar="0" padDir="LEFT" name="indassetref"/>  
4110 </option>  
4111 </level>  
4112 <level type="PURE\_IDENTITY" prefixMatch="urn:epc:id:giai" >  
4113 <option optionKey="12" pattern="urn:epc:id:giai:([0-9]{12})\.[0-9]{12}"  
4114 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >  
4115 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-  
4116 9]\*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>  
4117 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-  
4118 9]\*" length="12" padChar="0" padDir="LEFT" name="indassetref"/>  
4119 </option>  
4120 <option optionKey="11" pattern="urn:epc:id:giai:([0-9]{11})\.[0-9]{13}"  
4121 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >  
4122 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]\*"  
4123 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>  
4124 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-  
4125 9]\*" length="13" padChar="0" padDir="LEFT" name="indassetref"/>  
4126 </option>  
4127 <option optionKey="10" pattern="urn:epc:id:giai:([0-9]{10})\.[0-9]{14}"  
4128 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >  
4129 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]\*"  
4130 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>  
4131 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-  
4132 9]\*" length="14" padChar="0" padDir="LEFT" name="indassetref"/>  
4133 </option>  
4134 <option optionKey="9" pattern="urn:epc:id:giai:([0-9]{9})\.[0-9]{15}"  
4135 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >  
4136 <field seq="1" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]\*"  
4137 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>  
4138 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999999999" characterSet="[0-  
4139 9]\*" length="15" padChar="0" padDir="LEFT" name="indassetref"/>  
4140 </option>

```
4141 <option optionKey="8" pattern="urn:epc:id:giai:([0-9]{8})\.( [0-9]{16})"
4142 grammar="'urn:epc:id:giai:' companyprefix '.' indassetref" >
4143 <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
4144 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
4145 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999999999"
4146 characterSet="[0-9]*" length="16" padChar="0" padDir="LEFT" name="indassetref"/>
4147 </option>
4148 <option optionKey="7" pattern="urn:epc:id:giai:([0-9]{7})\.( [0-9]{17})"
4149 grammar="'urn:epc:id:giai:' companyprefix '.' indassetref" >
4150 <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
4151 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
4152 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999999999"
4153 characterSet="[0-9]*" length="17" padChar="0" padDir="LEFT" name="indassetref"/>
4154 </option>
4155 <option optionKey="6" pattern="urn:epc:id:giai:([0-9]{6})\.( [0-9]{18})"
4156 grammar="'urn:epc:id:giai:' companyprefix '.' indassetref" >
4157 <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
4158 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
4159 <field seq="2" decimalMinimum="0" decimalMaximum="99999999999999999"
4160 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="indassetref"/>
4161 </option>
4162 </level>
4163 <level type="LEGACY" prefixMatch="giai="
4164 requiredParsingParameters="companyprefixlength" >
4165 <option optionKey="12" pattern="giai=([0-9]{13,30})" grammar="'giai=' companyprefix
4166 indassetref" >
4167 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4168 characterSet="[0-9]*" name="giai"/>
4169 </option>
4170 <option optionKey="11" pattern="giai=([0-9]{12,30})" grammar="'giai=' companyprefix
4171 indassetref" >
4172 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4173 characterSet="[0-9]*" name="giai"/>
4174 </option>
4175 <option optionKey="10" pattern="giai=([0-9]{11,30})" grammar="'giai=' companyprefix
4176 indassetref" >
4177 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4178 characterSet="[0-9]*" name="giai"/>
4179 </option>
4180 <option optionKey="9" pattern="giai=([0-9]{10,30})" grammar="'giai=' companyprefix
4181 indassetref" >
4182 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4183 characterSet="[0-9]*" name="giai"/>
4184 </option>
4185 <option optionKey="8" pattern="giai=([0-9]{9,30})" grammar="'giai=' companyprefix
4186 indassetref" >
4187 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4188 characterSet="[0-9]*" name="giai"/>
4189 </option>
4190 <option optionKey="7" pattern="giai=([0-9]{8,30})" grammar="'giai=' companyprefix
4191 indassetref" >
4192 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4193 characterSet="[0-9]*" name="giai"/>
4194 </option>
4195 <option optionKey="6" pattern="giai=([0-9]{7,30})" grammar="'giai=' companyprefix
4196 indassetref" >
4197 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4198 characterSet="[0-9]*" name="giai"/>
4199 </option>
4200 <rule type="EXTRACT" inputFormat="STRING" seq="1" newFieldName="indassetref"
4201 characterSet="[0-9]*" function="SUBSTR(giai,companyprefixlength)"/>
4202 <rule type="EXTRACT" inputFormat="STRING" seq="2" newFieldName="companyprefix"
4203 characterSet="[0-9]*" function="SUBSTR(giai,0,companyprefixlength)"/>
4204 </level>
4205 </scheme>
4206 </epcTagDataTranslation>
```

4207 **8.11. GIAI-96 Global Individual Asset Identifier for 96-bit EPCs**

4208 <?xml version="1.0" encoding="UTF-8"??>

```

4209 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
4210 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
4211 xsi:schemaLocation="EpcTagDataTranslation.xsd">
4212   <scheme name="GIAI-96" optionKey="companyprefixlength" tagLength="96" >
4213     <level type="BINARY" prefixMatch="00110100"
4214     requiredFormattingParameters="filter,taglength" >
4215       <option optionKey="12" pattern="00110100([01]{3})000([01]{40})([01]{42})"
4216       grammar="'00110100' filter '000' companyprefix indassetref" >
4217         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4218         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4219         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
4220         bitLength="40" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
4221         <field seq="3" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
4222         bitLength="42" length="12" padChar="0" padDir="LEFT" name="indassetref"/>
4223       </option>
4224       <option optionKey="11" pattern="00110100([01]{3})001([01]{37})([01]{45})"
4225       grammar="'00110100' filter '001' companyprefix indassetref" >
4226         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4227         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4228         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
4229         bitLength="37" length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
4230         <field seq="3" decimalMinimum="0" decimalMaximum="999999999999"
4231         characterSet="[01]*" bitLength="45" length="13" padChar="0" padDir="LEFT"
4232         name="indassetref"/>
4233       </option>
4234       <option optionKey="10" pattern="00110100([01]{3})010([01]{34})([01]{48})"
4235       grammar="'00110100' filter '010' companyprefix indassetref" >
4236         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4237         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4238         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
4239         bitLength="34" length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
4240         <field seq="3" decimalMinimum="0" decimalMaximum="999999999999"
4241         characterSet="[01]*" bitLength="48" length="14" padChar="0" padDir="LEFT"
4242         name="indassetref"/>
4243       </option>
4244       <option optionKey="9" pattern="00110100([01]{3})011([01]{30})([01]{52})"
4245       grammar="'00110100' filter '011' companyprefix indassetref" >
4246         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4247         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4248         <field seq="2" decimalMinimum="0" decimalMaximum="9999999999" characterSet="[01]*"
4249         bitLength="30" length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
4250         <field seq="3" decimalMinimum="0" decimalMaximum="99999999999999"
4251         characterSet="[01]*" bitLength="52" length="15" padChar="0" padDir="LEFT"
4252         name="indassetref"/>
4253       </option>
4254       <option optionKey="8" pattern="00110100([01]{3})100([01]{27})([01]{55})"
4255       grammar="'00110100' filter '100' companyprefix indassetref" >
4256         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4257         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4258         <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[01]*"
4259         bitLength="27" length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
4260         <field seq="3" decimalMinimum="0" decimalMaximum="9999999999999999"
4261         characterSet="[01]*" bitLength="55" length="16" padChar="0" padDir="LEFT"
4262         name="indassetref"/>
4263       </option>
4264       <option optionKey="7" pattern="00110100([01]{3})101([01]{24})([01]{58})"
4265       grammar="'00110100' filter '101' companyprefix indassetref" >
4266         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4267         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4268         <field seq="2" decimalMinimum="0" decimalMaximum="9999999" characterSet="[01]*"
4269         bitLength="24" length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
4270         <field seq="3" decimalMinimum="0" decimalMaximum="9999999999999999"
4271         characterSet="[01]*" bitLength="58" length="17" padChar="0" padDir="LEFT"
4272         name="indassetref"/>
4273       </option>
4274       <option optionKey="6" pattern="00110100([01]{3})110([01]{20})([01]{62})"
4275       grammar="'00110100' filter '110' companyprefix indassetref" >
4276         <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[01]*"
4277         bitLength="3" length="1" padChar="0" padDir="LEFT" name="filter"/>
4278         <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[01]*"
4279         bitLength="20" length="6" padChar="0" padDir="LEFT" name="companyprefix"/>

```

```

4280         <field seq="3" decimalMinimum="0" decimalMaximum="999999999999999999"
4281 characterSet="[01]*" bitLength="62" length="18" padChar="0" padDir="LEFT"
4282 name="indassetref"/>
4283     </option>
4284 </level>
4285     <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:giai-96"
4286 requiredFormattingParameters="filter,taglength" >
4287         <option optionKey="12" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){12}]\.[(0-
4288 9){12})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4289             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4290 length="1" padChar="0" padDir="LEFT" name="filter"/>
4291             <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
4292 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
4293             <field seq="3" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
4294 9]*" length="12" padChar="0" padDir="LEFT" name="indassetref"/>
4295         </option>
4296         <option optionKey="11" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){11}]\.[(0-
4297 9){13})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4298             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4299 length="1" padChar="0" padDir="LEFT" name="filter"/>
4300             <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
4301 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
4302             <field seq="3" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-
4303 9]*" length="13" padChar="0" padDir="LEFT" name="indassetref"/>
4304         </option>
4305         <option optionKey="10" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){10}]\.[(0-
4306 9){14})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4307             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4308 length="1" padChar="0" padDir="LEFT" name="filter"/>
4309             <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
4310 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
4311             <field seq="3" decimalMinimum="0" decimalMaximum="9999999999999999" characterSet="[0-
4312 9]*" length="14" padChar="0" padDir="LEFT" name="indassetref"/>
4313         </option>
4314         <option optionKey="9" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){9}]\.[(0-
4315 9){15})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4316             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4317 length="1" padChar="0" padDir="LEFT" name="filter"/>
4318             <field seq="2" decimalMinimum="0" decimalMaximum="999999999999" characterSet="[0-9]*"
4319 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
4320             <field seq="3" decimalMinimum="0" decimalMaximum="9999999999999999" characterSet="[0-
4321 9]*" length="15" padChar="0" padDir="LEFT" name="indassetref"/>
4322         </option>
4323         <option optionKey="8" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){8}]\.[(0-
4324 9){16})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4325             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4326 length="1" padChar="0" padDir="LEFT" name="filter"/>
4327             <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
4328 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
4329             <field seq="3" decimalMinimum="0" decimalMaximum="9999999999999999"
4330 characterSet="[0-9]*" length="16" padChar="0" padDir="LEFT" name="indassetref"/>
4331         </option>
4332         <option optionKey="7" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){7}]\.[(0-
4333 9){17})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4334             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4335 length="1" padChar="0" padDir="LEFT" name="filter"/>
4336             <field seq="2" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
4337 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
4338             <field seq="3" decimalMinimum="0" decimalMaximum="99999999999999999999"
4339 characterSet="[0-9]*" length="17" padChar="0" padDir="LEFT" name="indassetref"/>
4340         </option>
4341         <option optionKey="6" pattern="urn:epc:tag:giai-96:([0-7]{1})\.[(0-9){6}]\.[(0-
4342 9){18})" grammar="urn:epc:tag:giai-96:' filter '.' companyprefix '.' indassetref" >
4343             <field seq="1" decimalMinimum="0" decimalMaximum="7" characterSet="[0-7]*"
4344 length="1" padChar="0" padDir="LEFT" name="filter"/>
4345             <field seq="2" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
4346 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
4347             <field seq="3" decimalMinimum="0" decimalMaximum="99999999999999999999"
4348 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="indassetref"/>
4349         </option>
4350     </level>

```



```
4351 <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:giai" >
4352 <option optionKey="12" pattern="urn:epc:id:giai:([0-9]{12})\.([0-9]{12})"
4353 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4354 <field seq="1" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-
4355 9]*" length="12" padChar="0" padDir="LEFT" name="companyprefix"/>
4356 <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-
4357 9]*" length="12" padChar="0" padDir="LEFT" name="indassetref"/>
4358 </option>
4359 <option optionKey="11" pattern="urn:epc:id:giai:([0-9]{11})\.([0-9]{13})"
4360 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4361 <field seq="1" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
4362 length="11" padChar="0" padDir="LEFT" name="companyprefix"/>
4363 <field seq="2" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-
4364 9]*" length="13" padChar="0" padDir="LEFT" name="indassetref"/>
4365 </option>
4366 <option optionKey="10" pattern="urn:epc:id:giai:([0-9]{10})\.([0-9]{14})"
4367 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4368 <field seq="1" decimalMinimum="0" decimalMaximum="99999999999" characterSet="[0-9]*"
4369 length="10" padChar="0" padDir="LEFT" name="companyprefix"/>
4370 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
4371 9]*" length="14" padChar="0" padDir="LEFT" name="indassetref"/>
4372 </option>
4373 <option optionKey="9" pattern="urn:epc:id:giai:([0-9]{9})\.([0-9]{15})"
4374 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4375 <field seq="1" decimalMinimum="0" decimalMaximum="999999999" characterSet="[0-9]*"
4376 length="9" padChar="0" padDir="LEFT" name="companyprefix"/>
4377 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999" characterSet="[0-
4378 9]*" length="15" padChar="0" padDir="LEFT" name="indassetref"/>
4379 </option>
4380 <option optionKey="8" pattern="urn:epc:id:giai:([0-9]{8})\.([0-9]{16})"
4381 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4382 <field seq="1" decimalMinimum="0" decimalMaximum="99999999" characterSet="[0-9]*"
4383 length="8" padChar="0" padDir="LEFT" name="companyprefix"/>
4384 <field seq="2" decimalMinimum="0" decimalMaximum="99999999999999999"
4385 characterSet="[0-9]*" length="16" padChar="0" padDir="LEFT" name="indassetref"/>
4386 </option>
4387 <option optionKey="7" pattern="urn:epc:id:giai:([0-9]{7})\.([0-9]{17})"
4388 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4389 <field seq="1" decimalMinimum="0" decimalMaximum="9999999" characterSet="[0-9]*"
4390 length="7" padChar="0" padDir="LEFT" name="companyprefix"/>
4391 <field seq="2" decimalMinimum="0" decimalMaximum="9999999999999999999"
4392 characterSet="[0-9]*" length="17" padChar="0" padDir="LEFT" name="indassetref"/>
4393 </option>
4394 <option optionKey="6" pattern="urn:epc:id:giai:([0-9]{6})\.([0-9]{18})"
4395 grammar="urn:epc:id:giai:' companyprefix '.' indassetref" >
4396 <field seq="1" decimalMinimum="0" decimalMaximum="999999" characterSet="[0-9]*"
4397 length="6" padChar="0" padDir="LEFT" name="companyprefix"/>
4398 <field seq="2" decimalMinimum="0" decimalMaximum="999999999999999999999"
4399 characterSet="[0-9]*" length="18" padChar="0" padDir="LEFT" name="indassetref"/>
4400 </option>
4401 </level>
4402 <level type="LEGACY" prefixMatch="giai="
4403 requiredParsingParameters="companyprefixlength" >
4404 <option optionKey="12" pattern="giai=([0-9]{13,30})" grammar="'giai=' companyprefix
4405 indassetref" >
4406 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4407 characterSet="[0-9]*" name="giai"/>
4408 </option>
4409 <option optionKey="11" pattern="giai=([0-9]{12,30})" grammar="'giai=' companyprefix
4410 indassetref" >
4411 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4412 characterSet="[0-9]*" name="giai"/>
4413 </option>
4414 <option optionKey="10" pattern="giai=([0-9]{11,30})" grammar="'giai=' companyprefix
4415 indassetref" >
4416 <field seq="1" decimalMinimum="0" decimalMaximum="9999999999999999999999999999999"
4417 characterSet="[0-9]*" name="giai"/>
4418 </option>
4419 <option optionKey="9" pattern="giai=([0-9]{10,30})" grammar="'giai=' companyprefix
4420 indassetref" >
```



```

4488     <field seq="1" characterSet="[0-9 A-HJ-NP-Z]*" name="cageordodaac"/>
4489     <field seq="2" decimalMinimum="0" decimalMaximum="16777215" characterSet="[0-9]*"
4490 name="serial"/>
4491     </option>
4492   </level>
4493 </scheme>
4494 </epcTagDataTranslation>
4495

```

## 4496 8.13. USDOD-96 DEPARTMENT OF DEFENCE for 96-bit 4497 EPCs

```

4498 <?xml version="1.0" encoding="UTF-8"?>
4499 <epcTagDataTranslation version="0.04" date="2005-04-18T16:05Z" epcTDSVersion="1.1r1.27"
4500 xmlns:xsi="http://www.w3.org/2001/XMLSchema"
4501 xsi:noNamespaceSchemaLocation="EpcTagDataTranslation.xsd">
4502   <scheme name="USDOD-96" optionKey="1" tagLength="96" >
4503     <level type="BINARY" prefixMatch="00101111" requiredFormattingParameters="taglength" >
4504       <option optionKey="1" pattern="00101111([01]{4})([01]{48})([01]{36})"
4505       grammar="'00101111' filter cageordodaac serial" >
4506         <field seq="1" decimalMinimum="0" decimalMaximum="15" characterSet="[01]*"
4507         bitLength="4" name="filter"/>
4508         <field seq="2" characterSet="[01]*" compaction="8-bit" padChar=" " padDir="LEFT"
4509         length="6" bitLength="48" name="cageordodaac"/>
4510         <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[01]*"
4511         bitLength="36" name="serial"/>
4512       </option>
4513     </level>
4514     <level type="TAG_ENCODING" prefixMatch="urn:epc:tag:usdod-96"
4515     requiredFormattingParameters="taglength" >
4516       <option optionKey="1" pattern="urn:epc:tag:usdod-96:([0-9])\.[0-9 A-HJ-NP-
4517       Z]{5,6})\.[0-9]*" grammar="'urn:epc:tag:usdod-96:' filter '.' cageordodaac '.' serial"
4518       >
4519         <field seq="1" decimalMinimum="0" decimalMaximum="15" characterSet="[0-9]*"
4520         name="filter"/>
4521         <field seq="2" characterSet="[0-9 A-HJ-NP-Z]*" name="cageordodaac"/>
4522         <field seq="3" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
4523         name="serial"/>
4524       </option>
4525     </level>
4526     <level type="PURE_IDENTITY" prefixMatch="urn:epc:id:usdod" >
4527       <option optionKey="1" pattern="urn:epc:id:usdod:([0-9 A-HJ-NP-Z]{5,6})\.[0-9]*"
4528       grammar="'urn:epc:id:usdod:' cageordodaac '.' serial" >
4529         <field seq="1" characterSet="[0-9 A-HJ-NP-Z]*" name="cageordodaac"/>
4530         <field seq="2" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
4531         name="serial"/>
4532       </option>
4533     </level>
4534     <level type="LEGACY" prefixMatch="cageordodaac" >
4535       <option optionKey="1" pattern="cageordodaac=([0-9 A-HJ-NP-Z]{5,6});serial=([0-9]*"
4536       grammar="'cageordodaac=' cageordodaac ';'serial=' serial" >
4537         <field seq="1" characterSet="[0-9 A-HJ-NP-Z]*" name="cageordodaac"/>
4538         <field seq="2" decimalMinimum="0" decimalMaximum="68719476735" characterSet="[0-9]*"
4539         name="serial"/>
4540       </option>
4541     </level>
4542 </scheme>
4543 </epcTagDataTranslation>

```

4544  
4545  
4546  
4547

## 9. Glossary (non-normative)

This section provides a non-normative summary of terms used within this specification. For normative definitions of these terms, please consult the relevant sections of the document.

Term	Meaning
[Numbering/Coding] Scheme	A well-defined method of assigning an identification code to an object / shipment / location / transaction
Serialised	Provides a unique serial number for each unique object referenced using that coding scheme
GTIN	Global Trade Item Number – used to identify traded objects and services.
SSCC	Serial Shipping Container Code – provides a globally unique reference number for each shipment
GLN	Global Location Number – used to identify physical locations but also legal and organizational entities and departments
GRAI	Global Returnable Asset Identifier – used to identify returnable assets such as pallets and crates, gas cylinders, etc.
GIAI	Global Individual Asset Identifier – used to identify assets owned by an organisation, which are not being traded – often used for tracking inventory of high value equipment
GID	General Identifier – original hierarchical structure proposed for EPC by Auto-ID Centre. GID is a generic scheme, not specifically aligned with any particular legacy coding scheme.
Levels of Representation	The way in which the identifier is represented. Examples of different types of representation include sequences of binary digits (bits), sequences of numeric or alphanumeric characters, as well as Uniform Resource Identifiers (URIs)
Input Value	The identifier to be translated. The format in which it is expressed is the Inbound Representation.
Inbound representation	The way in which the identifier is supplied to the translation software. This may be auto-detected from the input value.
Outbound representation	The way in which the output from the translation software should be expressed. This must be specified by the client.
Binary	A sequence of binary digits or bits, consisting of only the digits '0' or '1'

Term	Meaning
Non-Binary Form	An integer, numeric or alphanumeric character string when not expressed in the corresponding binary format
URI / URN	A Uniform Resource Identifier / Uniform Resource Name – a string that uniquely identifies any particular object. Unlike a URL (Uniform Resource Locator) which may change when a web page moves from one website to another, the URI is intended to be a permanent reference, fixed for all time – even if the underlying binding to a particular website address changes. The URI is therefore at a higher level of abstraction than a URL. Currently most web browser technology will only resolve URLs – but not URIs.
Tag-Encoding URI	A URI format which encodes the physical tag length and fast-filter values in addition to the information encoded in the pure-identity URI. Intended for low-level applications – e.g. sorting machines, tag writers, etc.
Pure-Identity URI	<p>A more abstract URI format that provides each object with a unique identity but conveys no information regarding the physical limitations of the tag used to deliver that EPC.</p> <p><i>If an object is tagged with either a 64-bit tag or a 96-bit tag, then although the binary representation and tag-encoding URIs will differ, the pure-identity URI will be the same. Intended for use by high-level applications which are not concerned with writing to tags nor sorting on packaging level.</i></p>
Legacy Coding	Existing numbering/coding schemes already in use. Examples include GTIN, SSCC, GLN, GRAI, GIAI from the GS1 family – but also many other numbering schemes, such as VIN (Vehicle Identification Number) – or those managed by IATA for the aviation industry or the US DOD for defence.
Physical Level[s]	Representations where the encoding conveys information about the physical tag length (number of bits) and/or the packaging/classification level of the object. Specifically, the binary representation and tag-encoding URI.
Identity Level[s]	Higher-level representations that say nothing about the physical tag length, nor include explicit information about the packaging/classification level. Specifically the pure-identity URI and legacy coding levels of representation
Supplied parameters	Parameters that shall be supplied in addition to the input value, mainly because the input value itself lacks specific information required for constructing the output.

Term	Meaning
Options	Variations to handle variable-length data partitions, such as those resulting from the variable-length Company Prefix in the GS1 family of coding schemes. Where multiple options are specified, the same number of options should be specified for each level of representation and translation should always translate from the matching option within the inbound level to the corresponding option within the outbound level.
Regular Expression Pattern	A notation for representing sub-patterns of particular groups of characters to match
ABNF Grammar	Augmented Backus-Naur Form. Defined in RFC 2234. [ <a href="http://www.ietf.org/rfc/rfc2234.txt">http://www.ietf.org/rfc/rfc2234.txt</a> ] Notation indicating how the result can be expressed through a concatenation of fixed literal values and values of variable fields, whose values are previously determined.
[Fast] Filter	A number which is used to conveniently select only EPCs of a particular packaging level or classification – e.g. a filter within a smart reader may be configured to report only the cases and pallets – but not all of the items within those cases. The fast filter value may also be used for filtering and sorting.
Header	A binary EPC prefix which indicates the coding scheme and usually also the tag length. Headers of 2 bits and 8 bits are defined in the EPC Tag Data Standards specification
Field	The variable elements of the EPC in any of its representations – each partition or field has a logical role, such as identifying the responsible company (e.g. the manufacturer of a trade item) or the object class or SKU. Tag Data Translation software uses the regular expression pattern to extract values for each field. These may be temporarily stored in variables or an associative array (key-value lookup table) until they are later required for substitution into the outbound format.

Term	Meaning
Rules	<p>There are already a number of requirements to perform various string manipulations and other calculations in order to comply with the current TDS specification. Neither the regular expression patterns nor the ABNF grammar contain any embedded inline functions. Instead, additional fields are embedded and a separate list of rules are provided, in order to define how their values should be derived from fields whose values are already known. The rules also indicate the context and running order in which they should be executed, namely by specifying the scheme, level and stage of execution (Extract or Format) and the running order as an integer index, with functions executed in ascending order of the sequence number indicated by the seq attribute</p>
Prefix Match	<p>The Prefix Match is a substring which is used to determine the scheme of the inbound string. This is merely a method of optimizing the performance of translation software by limiting the number of pattern-match tests that are required, since the translation software only attempts full pattern matching and processing for the options of those schemes/levels whose Prefix Match matches at the start of the input value.</p>
OptionKey	<p>The OptionKey is used to identify the appropriate option to use where multiple variations are specified to deal with partitions of variable length. A default strategy may be to simply iterate through all the possible options and find only one where the format string matches the inbound string. However, this approach fails when multiple options match the inbound value. In this case, the translation software can use the enumerated value of the OptionKey to select the appropriate option to use. Each option entry is numbered – and each level specifies (via the name of a field) the appropriate option to choose. For example for the GS1 codes, the level element always specifies that the OptionKey="companyprefixlength" , so for a Company Prefix of '0037000', then field "companyprefixlength" would be specified as 7 via the supplied parameters and therefore Option #7 would be chosen for both the inbound and outbound levels.</p>
Encoding	<p>A conversion process towards the binary representation, i.e in the direction:</p> <p>Legacy code → Pure-identity URI → Tag-encoding URI → Binary</p>

Term	Meaning
Decoding	<p>A conversion process away from the binary representation, i.e in the direction:</p> <p>Binary → Tag-encoding URI → Pure-identity URI → Legacy code → ONS hostname</p>
Built-In Functions	<p>Functions that should be supported by all implementations of the tag data translation software, irrespective of the programming language in which the software was actually written. See Table 6.</p>
TDT XML Markup	<p>A well-defined machine-readable structured packet of data that represents the patterns, grammar, rules, and field constraints for each identifier coding scheme. Tag data translation software should periodically receive updated versions or patches of the XML markup tables, which it can then use to update its own internal set of rules for performing the conversions, whether this is done at run-time or compile-time. We envisage that the XML Data Table should be freely downloadable and should ideally use human-readable tagnames. Furthermore, it should be possible to use XML transformation technologies such as XSLT to render it into a suitably formatted human-readable table for use in revised versions of the Tag Data Standards specification. Rendering the tables for the specification from the XML Data Table as the master table should avoid any inconsistencies being introduced between the TDS specification and the master table used by the translation software.</p>
[EPC] [Tag Data] Translation Software	<p>A piece of software that performs conversions between different representations of the EPC within any given coding scheme. The translation software may be a library module or object which may be accessed by / embedded within any technology component in the EPC Network technology stack. It may also be implemented as a standalone service, such as an interactive web page form or a web service for automated batch-processing of conversions.</p>
EPC Tag Data Validation Software	<p>Software which need not perform any translation but may nevertheless make use of the Tag Data Translation markup files in order to validate that an EPC in any of its representations conforms to a valid format.</p>



Term	Meaning
EPC Network [Technology] Stack	<p>This consists of several architectural building blocks in order to connect physical objects with information systems. The technology stack includes:</p> <p>EPC – the Electronic Product Code</p> <p>Tags and Readers</p> <p>Filtering and Collection middleware</p> <p>Object Name Service (ONS)</p> <p>EPC Information Service (EPCIS).</p>
Checksum / Check Digit	<p>A number that is computed algorithmically from other digits in a numerical code in order to perform a very basic check of the integrity of the number; if the check digit supplied does not correspond to the check digit calculated from the other digits, then the number may have been corrupted. The check digit is in a way analogous to a message digest of a data packet or software package – except that message digests tend to be more robust since they consist of strings of several characters and hence many more possible permutations than a single check digit 0-9, with the result that there is a much smaller probability that a corrupted number or data packet will product the same message digest than that it will fortuitously produce a valid check digit. The algorithm for computing the check digit for EAN.UCC GS1 coding schemes is specified at <a href="http://www.ean-int.org/cdcalcul.html">http://www.ean-int.org/cdcalcul.html</a></p> <p>ISO 7064 is a standard specifying a generic framework for check digit calculations.</p>
[EAN.UCC] Company Prefix	<p>A number allocated by the EAN.UCC which uniquely specifies a unique company – often the manufacturer of a trade item</p>
Company Prefix Index	<p>An integer used to obtain the full EAN.UCC Company Prefix via a lookup table, keyed on the smaller integer number of the Company Prefix Index. This is used with the 64-bit schemes in order to allocate a larger range of bits for the remaining data partitions. The Company Prefix Index is tabulated in XML and comma-separated value formats at <a href="http://www.onsepc.com">http://www.onsepc.com</a></p>

4548

4549 **10. References**

4550

4551 TDS - EPCglobal Tag Data Standards

4552 ONS - Object Name Service

4553 See <http://www.epcglobalinc.org/under 'Standards'>

4554

4555 GTIN – Global Trade Item Number

4556 GLN – Global Location Number

4557 SSCC – Serial Shipping Container Code

4558 GRAI – Global Returnable Asset Identifier

4559 GIAI – Global Individual Asset Identifier

4560 GS1 (formerly EAN UCC Company Prefix)

4561 GS1 Check Digit Calculation

4562 See <http://www.gs1.org> under 'The EAN.UCC System' > 'Identification'

4563

4564 US DOD / CAGE and DODAAC codes in passive tags

4565 See <http://www.acq.osd.mil/log/rfid/> under 'Passive Tag Data'

4566

4567 NAPTR – Naming Authority Pointer records

4568 See RFC2915 at <http://www.ietf.org/rfc/rfc2915.txt?number=2915>

4569

4570 PCRE – Perl-Compliant Regular Expressions

4571 See <http://www.pcre.org>

4572

4573 ABNF – Augmented Backus-Naur Form

4574 See RFC2234 at <http://www.ietf.org/rfc/rfc2234.txt?number=2234>

4575

4576 URI – Uniform Resource Identifiers

4577 See RFC2396 at <http://www.ietf.org/rfc/rfc2396.txt?number=2234>

4578

4579 CGI – Common Gateway Interface

4580 See <http://hoohoo.ncsa.uiuc.edu/cgi/>

- 4581
- 4582 UML – Unified Modelling Language
- 4583       See <http://www.uml.org/>
- 4584
- 4585 ISO AFI – Application Family Identifier
- 4586       See ISO/IEC 15693 and ISO/IEC 15961 and 15962
- 4587
- 4588 EPCglobal UHF Generation 2 Protocol
- 4589       See <http://www.epcglobalinc.org>
- 4590
- 4591 XML DOM (Document Object Model) and XPath
- 4592       See <http://www.w3.org/TR/xpath>