DoD Tag Data Constructs

Fred Rowe

Tag Data Constructs - Q & A

Q. What is a DoD tag data construct?
A. A systematic method of encoding a unique number (EPC) on an RFID tag that will uniquely identify a pallet or case of goods to be shipped to DoD.

Q. How will I know that my RFID tag numbers are unique?
A. As detailed in this presentation, you will use your CAGE code to create the unique number for the RFID tag.

Q. Do I have to join EPCglobal in order to be able to encode RFID tags for use in shipping to DoD?
A. No. You may choose to do so, but it is not required.

Q. If I have questions regarding RFID tagging for DoD what document can I consult?
A. The DoD Suppliers' Passive RFID Information Guide which can found at: www.dodrfid.org/supplierguide.htm

Tag Data Constructs – CAGE code

CAGE code:

- Commercial and Government Entity Code. A five character alphanumeric identification system administered by the Defense Logistics Systems Center (DLSC) in Battle Creek, Michigan, and used in supply management.

- "… a five position code that identifies companies doing or wishing to do business with the Federal Government. The format of the code is the first and fifth position must be numeric. The second, third and fourth may be any mixture of alpha/numeric excluding I (eye) and O (oh) …"

- If you are a current supplier to DoD, you should already have a CAGE that can be used to create an properly encoded RFID tag.

Tag Data Constructs – DoD-64

This tag data construct should be used to encode 64-bit Class 0 and Class 1 tags for shipping goods to the DoD. The 64-bit tag is broken into fields as indicated below:

Header	Filter	Government Managed Identifier (CAGE)	Serial Number
8 bits	2 bits	30 bits	24 bits

Header – specifies that the tag data is encoded as a DoD 64-bit tag construct, use binary number 1100 1110

Filter – identifies a pallet, case, or UID item associated with tag, represented in binary number format using the following values:

- 00 = pallet
- 01 = case
- 10 = UID item
- 11 = reserved for future use

Tag Data Constructs – DoD-64 (cont.)

Header	Filter	Government Managed Identifier (CAGE)	Serial Number
8 bits	2 bits	30 bits	24 bits

Government Managed Identifier – For suppliers, this field will be encoded with their CAGE code, identifying the supplier and ensuring uniqueness of serial number across all suppliers, and is represented in truncated ASCII format. This format truncates the two most significant bits of the standard 8-bit ASCII representation of the characters of the CAGE code. Once truncated, the remaining 6 bits still uniquely identify the original ASCII characters and can be properly decoded after the encoding scheme.

Serial Number – uniquely identifies up to 2^{24} = 16,777,216 tagged items, represented in binary number format. After the serial number is converted into binary format, it must be left padded with zeros to 24 bits total. It is the responsibility of the supplier to insure that this is a unique number across all shipments to DoD.

Tag Data Constructs – DoD-64 example

Encoding example: A supplier with **CAGE code 1D381** wishes to encode a **64-bit tag** for use on a **case** of goods that is uniquely identified with the number **16,522,293**

Step 1 Select DoD header value based on tag size

Header

1	1	0	0	1	1	1	0

Header type	Value
DoD-64	**11001110**
DoD-96	11001111

Step 2 Select filter value based on object being tagged

Filter

0	1

Object type	Value
Pallet	00
Case	**01**
UID item	10
Reserved	11

Tag Data Constructs – DoD-64 example

Step 3 Encode CAGE chars using mapping table values

CAGE code

$$\underbrace{1\,1\,0\,0\,0\,1}_{\text{‘1’}}\;\underbrace{0\,0\,0\,1\,0\,0}_{\text{‘D’}}\;\underbrace{1\,1\,0\,0\,1\,1}_{\text{‘3’}}\;\underbrace{1\,1\,1\,0\,0\,0}_{\text{‘8’}}\;\underbrace{1\,1\,0\,0\,0\,1}_{\text{‘1’}}$$

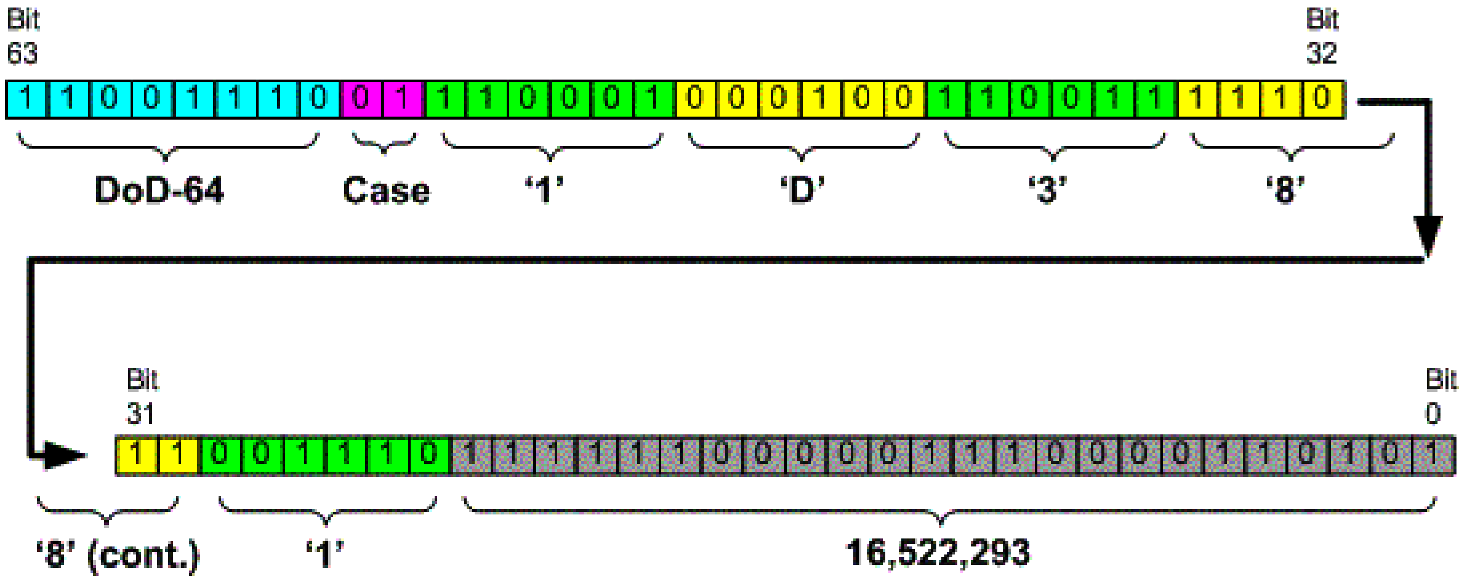
Step 4 Encode unique serial number

Serial number

$$\underbrace{1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0\,0\,0\,1\,1\,0\,1\,0\,1}_{16{,}522{,}293}$$

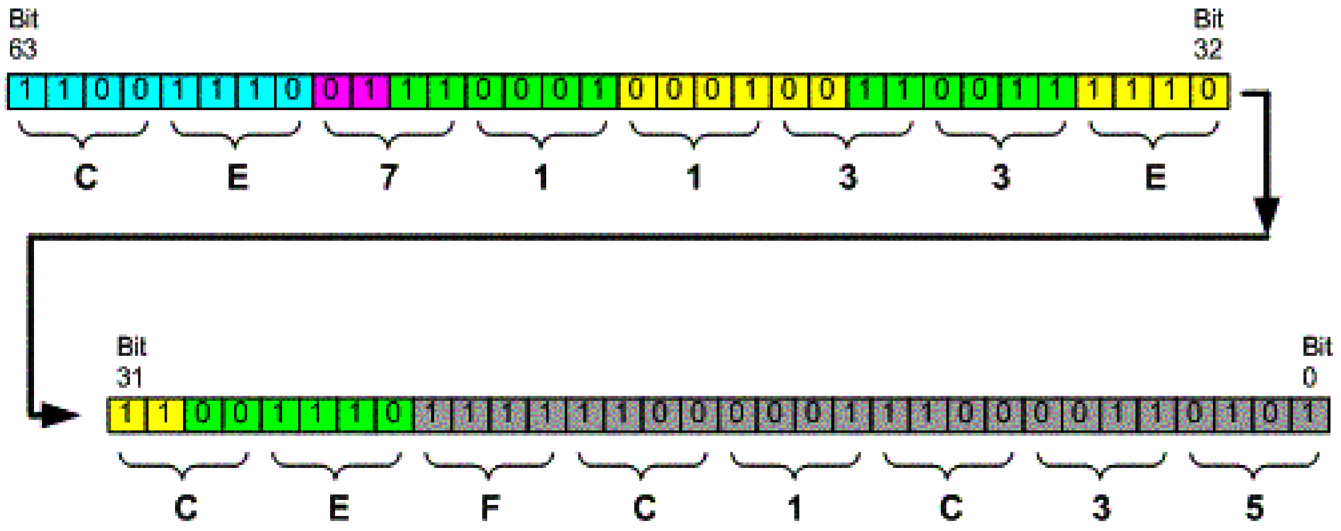
Tag Data Constructs – DoD-64 example

Step 5 Place the fields into the proper order within the 64-bits of the tag



Tag Data Constructs – DoD-64 example

Step 6 **Convert the 64-bit binary (base 2) number into hexadecimal (base 16) format for encoding**



The result is a unique number expressed in hexadecimal format that can be written to the tag: **CE71133ECEFC1C35.** This is the same number that must be communicated in an ASN EDI document via WAWF.

Tag Data Constructs – DoD-96

This tag data construct should be used to encode 96-bit Class 0 and Class 1 tags for shipping goods to the DoD. The 96-bit tag is broken into a number of fields as indicated below:

Header	Filter	Government Managed Identifier (CAGE)	Serial Number
8 bits	4 bits	48 bits	36 bits

Header – specifies that the tag data is encoded as a DoD 96-bit tag construct, use binary number 1100 1111

Filter – identifies a pallet, case, or UID item associated with tag, represented in binary number format using the following values:

- 0000 = pallet
- 0001 = case
- 0010 = UID item
- 0011 = reserved for future use

Tag Data Constructs – DoD-96 (cont.)

Header	Filter	Government Managed Identifier (CAGE)	Serial Number
8 bits	4 bits	48 bits	36 bits

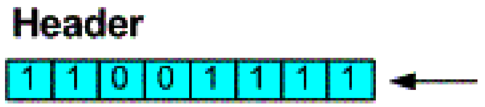
Government Managed Identifier – For suppliers, this field will be encoded with their CAGE code. This code identifies the supplier and ensures uniqueness of serial number across all suppliers, and is represented in standard 8-bit ASCII format. For the DoD-96 tag data construct, an ASCII space char must be prepended to the CAGE code to make the code a total of 6 ASCII chars.

Serial Number – uniquely identifies up to 2^{36} = 68,719,476,736 tagged items, represented in binary number format. After the serial number is converted into binary format, it must be left padded with zeros to 36 bits total.

Tag Data Constructs – DoD-96 example

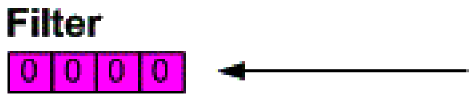
Encoding example: A supplier with **CAGE code 2S194** wishes to encode a **96-bit tag** for use on a **pallet** of goods that is uniquely identified with the number **12,345,678,901**

Step 1 **Select DoD header value based on tag size**



Header type	Value
DoD-64	11001110
DoD-96	**11001111**

Step 2 **Select filter value based on object being tagged**

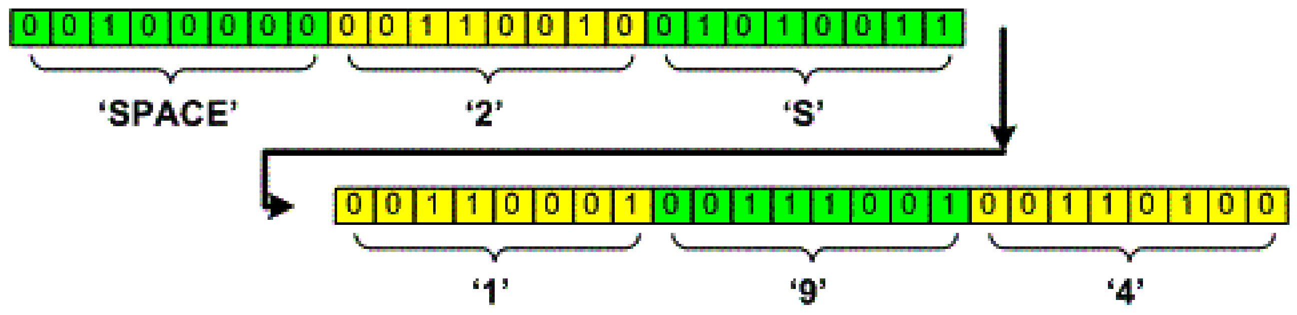


Object type	Value
Pallet	**0000**
Case	0001
UID item	0010
Reserved	0011 - 1111

Tag Data Constructs – DoD-96 example

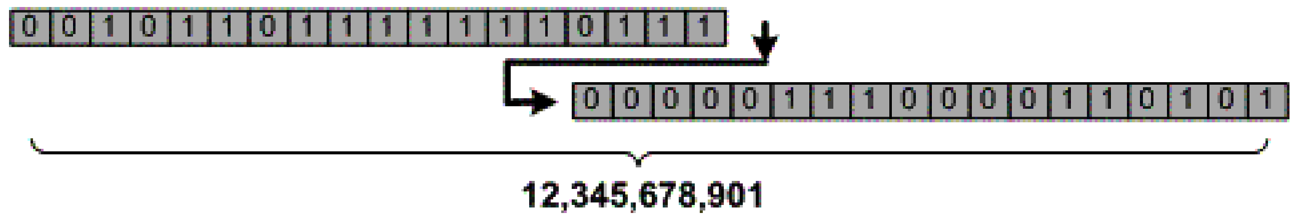
Step 3 Encode CAGE using ASCII values, remembering to pad a SPACE char on the left side

CAGE code



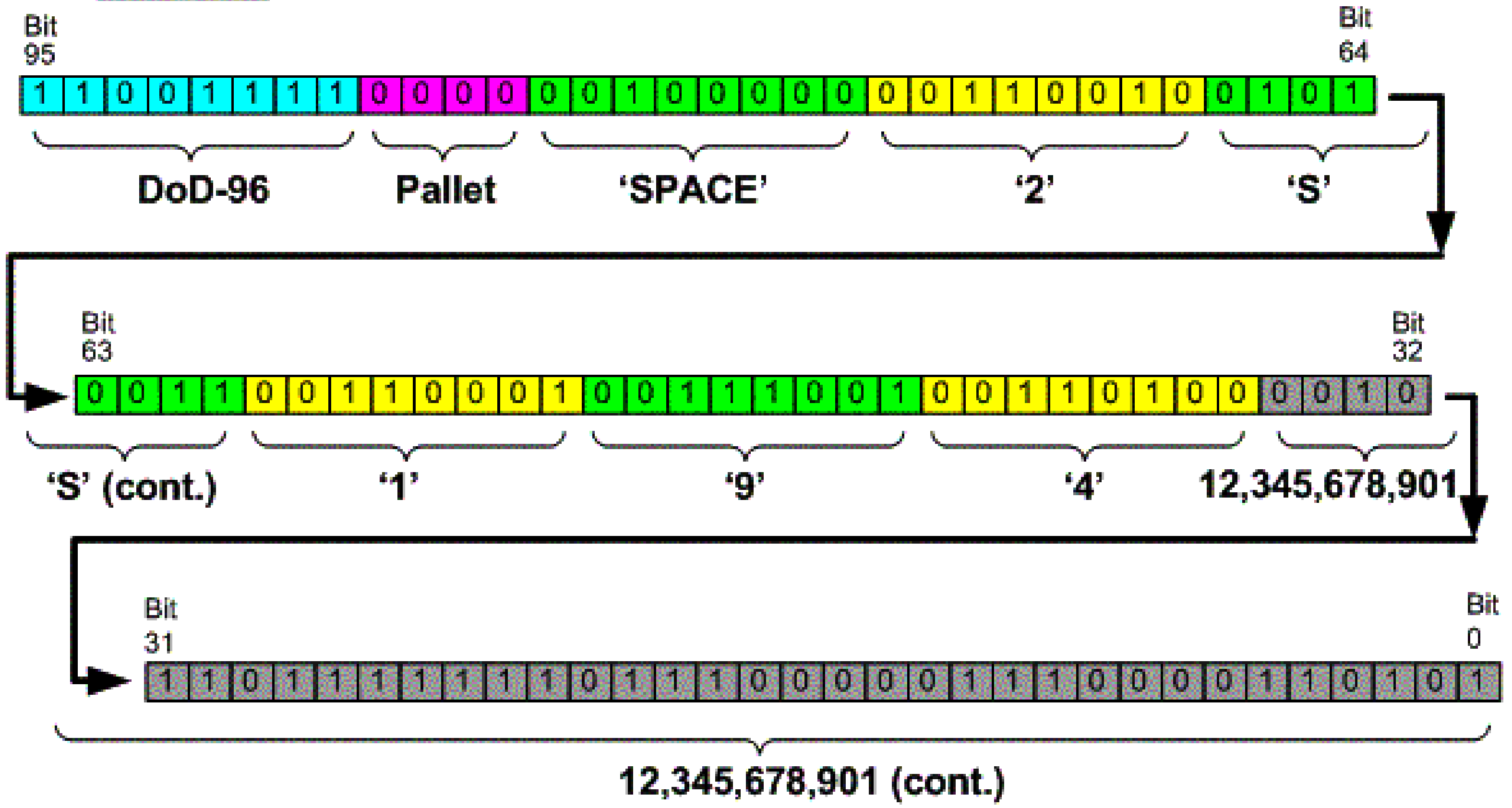
Step 4 Encode unique serial number

Serial number



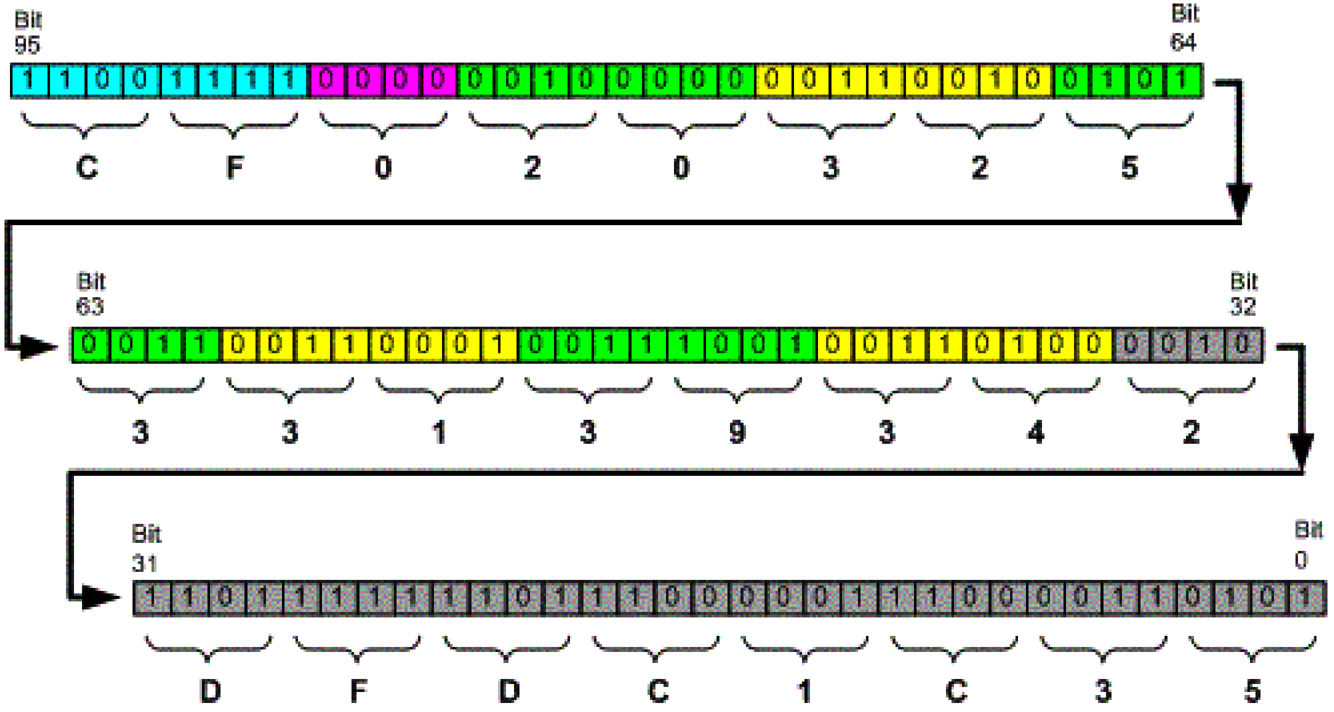
Tag Data Constructs – DoD-96 example

Step 5 Place the fields into the proper order within the 96-bits of the tag



Tag Data Constructs – DoD-96 example

Step 6 Convert the 96-bit binary (base 2) number into hexadecimal (base 16) format for encoding



The result is a unique number expressed in hexadecimal format that can be written to the tag: **CF020325331393342DFDC1C35.** This is the same number that must be communicated in an ASN EDI document via WAWF.