

### Background Information

Many vendors already ship items with EPC tags. These UPCs conform to the GS1 standards for UPC creation and assign a Global Trade Item Number (GTIN), therefore, Target's systems must be able to read these and map the encoded numbers back to an internal Target Corporation Item Number (TCIN) and to possibly many Department Class Item numbers (DPCI). When a serial number is added to a GTIN to uniquely identify the item, it is known as a Serialized GTIN or SGTIN. The GTIN includes:

- Company Prefix: a variable 6 to 12 digit company prefix of the assigning company
- Item Reference: a 7 to 1 digit Item reference and indicator
- Check Digit: 1 digit for barcode check digit (this need not be encoded in EPC tags)

These can be encoded in a 14 digit GS1 barcode. Any 12 digit GTIN used in the US and Canada can be converted to a 14 digit GTIN by prefixing it with two zeros. The first zero is an indicator digit meaning an individual item for retail sale, and the second is the country/region code for the US and Canada. If an item is only meant for sale in this region, these digits can be omitted and the GTIN 12 used.

Vendor Generated UPC Examples:

URI	GTIN 12	GTIN 14
urn:epc:id:sgtin:0038000.035621.12345	038000356216	00038000356216
urn:epc:id:sgtin:0030878.099916.67890	030878999168	00030878999168
urn:epc:id:sgtin:0085387.030438.12345	085387304385	00085387304385

### Vendor Generated EPC Schema

Vendor generated UPCs for source tagged product will follow the GS1 SGTIN format

[http://www.gs1.org/sites/default/files/docs/epc/TDS\\_1\\_9\\_Standard.pdf](http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf)

**6.3.1. Serialized Global Trade Item Number (SGTIN):** Page 26

Vendor Generated UPC Encoding Examples:

Barcode	Company Prefix	ObjectClass (No Check Digit)	Serial Number	SGTIN Hex
00038000356216	0038000	035621	12345	30340251C022C94000003039
030878999168	0030878	099916	67890	303401E27861930000010932
085387304385	0085387	030438	12345	303405362C1DB98000003039

**Note:** see <http://www.kentraub.net/tools/tagxlate/EPCEncoderDecoder.html> for a good tool to validate encodings (use GTIN14 with filter value of 1 – POS Item).

#### Background Information

Many of Target's vendors generate a 49 UPC. Since this is not a globally unique GTIN, an alternate encoding can be used for this 12 digit barcode:

- Positions 1 and 2: 49
- Positions 3-11: Target's internal financial/merchandising categorization and identification number (DPCI). DPCI is
  - 3 digit Department
  - 2 digit Class
  - 4 digit Item Number
- Position 12: system calculated Check Digit

Target Generated 49 UPC Examples:

URI	DPCI	Barcode
urn:epc:id:gid:4928300.16670.5000000001	283-00-1667	492830016670
urn:epc:id:gid:4902908.8545.5000000002	029-08-0854	490290808545
urn:epc:id:gid:4902907.15317.5000000003	029-07-1531	490290715317

#### Target Generated 49 EPC Schema

Target's 49 UPC source tagged product will follow a variation of the GS1 GID format

[http://www.gs1.org/sites/default/files/docs/epc/TDS\\_1\\_9\\_Standard.pdf](http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf)

**6.3.11. General Identifier (GID):** Page 31

Target will follow this construct:

- ManagerNumber: 49 + 3 digit Department + 2 digit Class (always starts with '49')
- ObjectClass: 4 digit Item Number + Check Digit (add leading 0's till 5 digits)
- SerialNumber: 11 digit, serialized unique ID (add leading 0's till 11 digits, max 68719476735)

Target uniquely assigns the leading 2 digits of the Serial # to each tag supplier ("Commissioning Authority"). Please consult a Target representative for your Commissioning Authority ID. Decode the serial number first and reference slide 4 if it begins with **01, 03 or 04**.

Target Generated 49 UPC Encoding Examples:

Barcode	Manager Number	ObjectClass (w/Check Digit)	Serial Number	GID Hex (drop leading zeros before encoding)
492830016670	4928300	16670	05000000001	3504B332C00411E12A05F201
490290808545	4902908	08545	05000000002	3504ACFFC00216112A05F202
490290715317	4902907	15317	05000000003	3504ACFFB003BD512A05F203

**Important note:** Using the 49 code and department class for the manager number, not a GS1 assigned Company Prefix, means this is not guaranteed to be globally unique.

### Deprecated – Do NOT USE Going Forward

#### Background Information

Some vendors ship items with EPC tags that encode our target 49 barcodes with the GS1 standard SGTIN format. These SGTINs are not global GTINs, but it is acceptable GS1 practice to use barcodes that begin with 49 for internal use. These SGTINs can be unpacked to get the associated DPCI:

- Positions 1 and 2: 49
- Positions 3-11: Target's internal financial/merchandising categorization and identification number (DPCI). DPCI is
  - 3 digit Department
  - 2 digit Class
  - 4 digit Item Number
- Position 12: system calculated Check Digit

These can be encoded in a 14 digit GS1 barcode. Any 12 digit GTIN used in the US and Canada can be converted to a 14 digit GTIN by prefixing it with two zeros. The first zero is an indicator digit meaning an individual item for retail sale, and the second is the country/region code for the US and Canada. If an item is only meant for sale in this region, these digits can be omitted and the GTIN 12 used.

Target Generated 49 UPC Examples:

URI	GTIN 12	GTIN 14
urn:epc:id:sgtin:0498000.035621.12345	498000356216	00498000356216
urn:epc:id:sgtin:0490878.099916.67890	490878999168	00490878999168
urn:epc:id:sgtin:0495387.030438.12345	495387304385	00495387304385

#### Target Generated 49 EPC Schema (Alternate – NOT preferred)

Target's 49 UPC source tagged product may also follow the GS1 SGTIN format

[http://www.gs1.org/sites/default/files/docs/epc/TDS\\_1\\_9\\_Standard.pdf](http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf)

**6.3.1. Serialized Global Trade Item Number (SGTIN):** Page 26

Target Generated 49 UPC Encoding Examples (Alternate):

Barcode	Company Prefix	ObjectClass (No Check Digit)	Serial Number	SGTIN Hex
00498000356216	0498000	035621	12345	30341E6540037A8000003039
490878999168	0490878	099916	67890	30341DF5F861930000010932
495387304385	0495387	030438	12345	30341E3C6C1DB98000003039

**Note:** see <http://www.kentraub.net/tools/tagxlate/EPCEncoderDecoder.html> for a good tool to validate encodings (use GTIN14 with filter value of 1 – POS Item).

#### Background Information

If a tag is lost (either for returned product, or missing from a sales floor item), the Store team will provision a replacement tag. For vendor generated UPCs, we won't be able to assure a unique serial number to create a new SGTIN. As we are the provisioning authority for our 49 UPCs, we can provision a replacement tag that will be unique. Both vendor generated UPC and Target generated 49 UPC replacement tags will be based on the original UPC code to be replaced:

- UPC: 12 digit barcode (for Target generated UPCs, this starts with 49)
- GTIN: 00 + barcode

Examples (original, not replacement):

URI	UPC/GID
urn:epc:id:sgtin:0038000.035621.12345	00038000356216
urn:epc:id:sgtin:0030878.099916.67890	00030878999168
urn:epc:id:gid:4928300.16670.1000000001	00492830016670

#### In Store Retagging in GID EPC Schema

In store replacement tags for both vendor generated UPCs and and Target generated 49 UPCs will follow a variation of the GS1 GID format

[http://www.gs1.org/sites/default/files/docs/epc/TDS\\_1\\_9\\_Standard.pdf](http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf)

**6.3.11. General Identifier (GID):** Page 31

Target will follow this construct:

- ManagerNumber: Digits 1 through 8 of UPC (add leading 0's till 8 digits)
- ObjectClass: Digits 9 through 14 of UPC (add leading 0's till 6 digits, includes check digit)
- SerialNumber: 11 digit, serialized unique ID (add leading 0's till 11 digits, max 68719476735)

Target uses one of the reserved Commissioning Authority numbers (first two digits of serial number) for these tags: **01, 03 or 04**. Decode the serial number first and reference slide 2 if it **DOESN'T** begin with **01, 03 or 04**.

Encoded Replacement Tag Examples:

Barcode	Manager Number	ObjectClass (w/Check Digit)	Serial Number	GID Hex (drop leading zeros before encoding)
00038000356216	<b>000</b> 38000	356216	<b>0</b> 4000000001	350009470056F780EE6B2801
030878999168	<b>000</b> 30878	999168	<b>0</b> 4000000002	35000789E0F3F000EE6B2802
492830016670	<b>00</b> 492830	<b>0</b> 16670	<b>0</b> 4000000003	35007851E00411E0EE6B2803

**Important note:** Using the shifted UPC (GTIN) for the manager number, not a GS1 assigned Company Prefix, means this is not guaranteed to be globally unique.



ID	Authority	Encode/Decode	Notes
00	Reserved TGT Internal	Slide 2	No Use Yet
<b>01</b>	Reserved TGT Internal	<b>Slide 4</b>	Retagging - <b>No Use Yet</b>
02	<b>Checkpoint</b>	Slide 2	Source/Supply Chain/In Store Tagging
<b>03</b>	Reserved TGT Internal	<b>Slide 4</b>	Retagging - <b>No Use Yet</b>
<b>04</b>	Reserved TGT Internal	<b>Slide 4</b>	Retagging - <b>No Use Yet</b>
05	<b>Avery Dennison</b>	Slide 2	Source Tagging
06	<b>SML</b>	Slide 2	Source Tagging
07	<b>SML</b>	Slide 2	Source Tagging
08	<b>Checkpoint</b>	Slide 2	Source Tagging - No Use Yet
09	[Open]		
10	[Open]		
11	[Open]		
12	[Open]		
13	[Open]		
14	Ceva	Slide 2	Source Tagging
...	[Open]		
67	[Open]		This is the maximum available

**NOTE:** None of the reserved retagging IDs on **Slide 4** are currently used (**01, 03, 04**).

NOTE: Max ID is 67, as serial numbers beginning with 68 for GID encodings can only go up to 68719476735 due to limitations in the number of bits in the binary encoding.

### Background Information

Target's modern, omnichannel supply chain leverages a common Target Corporation Item Number (TCIN) to identify all retail items. This is a random 10 digit number assigned by our new Product Information Management system (PIM) and is currently used by our online properties and modern supply chain systems. The TCIN identifies an item at a more granular level than the historical DPCI, and allows for correct identification of the exact product that the guest desires (e.g. Batman vs. Superman costume, which may be the same DPCI).

A TCIN is random with no inherent logic so the number itself cannot identify the item without referring to the product catalog. The TCIN is also never represented in barcode format so must be mapped to associated UPC barcodes.

Example TCINs:

URI	Description	TCIN
urn:epc:id:tcin:0013951442.100000000000001	Waterpik DSL-653	13951442
urn:epc:id:tcin:0014448.79610000000000002	Oster DuraCeremantic Waffle Iron	14448796
urn:epc:id:tcin:0016399080.100000000000003	GoPro Hero4 Silver Edition	16399080

### Target Common Item Number TCIN Schema

Target will define a TCIN-96 encoding similar to the GS1 Global Individual Asset Identifier GIAI, but will use a reserved header: 0000 1000.

Target TCIN-96 encoding will follow this construct:

- TCIN: 10 digit TCIN (add leading 0's till 10 digits)
- SerialNumber: 15 digit, serialized unique ID (add leading 0's to serial number until 15 digits)

The TCIN makes this unique, so the TCIN field is longer in this scheme.

Target will provide a TCIN-96 service (RESTful API) that will generate a guaranteed unique EPC number for any provided TCIN. Contact the Target RFID team for more information.

Target Examples:

TCIN	Padded TCIN	Padded Serial Number	TCIN-96 Hex
13951442	0013951442	123456789012345	0800035387487048860DDF79
14448796	0014448796	543210987654321	08000371E271EE0C29F50CB1
16399080	0016399080	135792468012345	080003E8EBA07B80A5D1AD39

**Important note:** This is not a GS1 compliant encoding, and is only intended for use within Target's systems.

#### Background Information

Target tags and tracks many non retail items. These may be tracked independently away from the retail floor (e.g. trailers), but some may also be comingled in the retail space. It is therefore imperative that the EPC encodings conform to a set standard for uniqueness so as to not interfere or overlap with the EPC encodings of the retail items (e.g. a shelf tag next to a tagged retail product can both be read and not interfere with each other). There are many categories of non-retail items and the complete table of those supported begins on slide 9.

The many and varied asset tracking systems in use at Target generate their own unique asset IDs (e.g. shelf IDs or Team Member handhelds). An ideal solution should directly leverage these unique Asset IDs and not replace them or add a cross reference.

Example TIAIs:

URI	Description	Asset Ref	Asset ID
urn:epc:id:tiaia:005.12345678901234	Team Member Device	005	12345678901234
urn:epc:id:tiaia:007.Y2A630	SPARK ID (PD&D)	007	Y2A630
urn:epc:id:tiaia:008.929B35AD6B1BA6B5	IoT LoRaWAN ID (64 bit number)	008	929B35AD6B1BA6B5

#### Target Individual Asset Identifier TIAI-A Schema

Target will define a TIAI-A-96 encoding similar to the GS1 Global Individual Asset Identifier GIAI, but will use a reserved header: 0000 1010

Target will follow this construct:

- Asset Ref: 3 digit number (add leading 0's till 3 digits)
- Asset ID: 21 digit (or 12 varchar) unique ID (add leading 0's till 21 digits)

The Asset ID makes this unique, so the Asset ID field is longer in this scheme.

See the 6 bit decoding reference on slide 8 if the Asset ID needs to be decoded to a alphanumeric varchar to reference back to the Asset Reference system.

Target Examples:

Asset Ref	Asset ID	TIAI-96 Hex
005	12345678901234	0A00050000000B3A73CE2FF2
007	Y2A630 (see next slide for encoding example)	0A0007000000000672076CB0
008	929B35AD6B1BA6B5 (note, hex representation of 64 binary bits)	0A000800929B35AD6B1BA6B5

**Important note:** This is not a GS1 compliant encoding, and is only intended for use within Target's systems. Also note if the Asset ID is longer than 21 digits or 12 characters, a cross reference table will be necessary for that system, and the system will have to handle that.



## 6 Bit Encoding/Decoding Reference

## Target Internal Use Only

A TIAI leverages a 72 bit, unique Asset Identifier field. This field is unique for the associated Asset Reference, so if you are using the entire encoded number for comparison, you may never need to decode it (hex, binary, decimal, it doesn't matter, all are unique). However, sometimes you want to decode it to tie it back to the Asset Reference system which generated it. If it is an integer, you can decode it with a standard binary to decimal converter. If it is an alphanumeric varchar, you will need to use a 6 bit decoding table.

Graphic Symbol	Name	Binary Value	URI Form	Graphic Symbol	Name	Binary Value	URI Form
#	Pound/Number Sign	100011	%23	H	Capital H	001000	H
-	Hyphen/Minus Sign	101101	-	I	Capital I	001001	I
/	Forward Slash	101111	%2F	J	Capital J	001010	J
0	Zero Digit	110000	0	K	Capital K	001011	K
1	One Digit	110001	1	L	Capital L	001100	L
2	Two Digit	110010	2	M	Capital M	001101	M
3	Three Digit	110011	3	N	Capital N	001110	N
4	Four Digit	110100	4	O	Capital O	001111	O
5	Five Digit	110101	5	P	Capital P	010000	P
6	Six Digit	110110	6	Q	Capital Q	010001	Q
7	Seven Digit	110111	7	R	Capital R	010010	R
8	Eight Digit	111000	8	S	Capital S	010011	S
9	Nine Digit	111001	9	T	Capital T	010100	T
A	Capital A	000001	A	U	Capital U	010101	U
B	Capital B	000010	B	V	Capital V	010110	V
C	Capital C	000011	C	W	Capital W	010111	W
D	Capital D	000100	D	X	Capital X	011000	X
E	Capital E	000101	E	Y	Capital Y	011001	Y
F	Capital F	000110	F	Z	Capital Z	011010	Z
G	Capital G	000111	G	<NULL>	Blank Placeholder	000000	<NULL>

[illegible]

8





Asset Ref	Tracking System	Asset ID	Notes
001	Reserved	Reserved	No Use Yet
002	Reserved	Reserved	No Use Yet
003	Reserved	Reserved	No Use Yet
004	Reserved	Reserved	No Use Yet
005	Zebra	14 digit SN number	Zebra Team Member Device
006	Grovesite	6 digit decimal	PD&D Sample ID (Old Tracking System)
007	SPARK	4+ varchar	PD&D Sample ID (New Tracking System)
008	IoT	64 bit LoRaWAN	IoT Platform Team Tracked Device
009	IoT	<b>UUID (TBD)</b>	IoT Platform Team Tracked Device
010	[Open]		No Use Yet
011	[Open]		No Use Yet
012	[Open]		No Use Yet
013	[Open]		No Use Yet
014	[Open]		No Use Yet
015	[Open]		No Use Yet
016	[Open]		No Use Yet
...	[Open]		No Use Yet
8191	[Open]		This is the maximum available

**NOTE:** Max Asset Ref is 8191, so guaranteed 3 digits, but almost 4 digits, due to limitations in the number of bits in the binary encoding. If more are needed, another header can be added.