Published June 1, 2003. Distribution restricted to Sponsors until September 1, 2003.

AUTO-ID CENTER

WHITE PAPER

EPC™ Tag Data Specification

David Brock, Chris Cummins

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 400 TECHNOLOGY SQ, BUILDING NE46, 6TH FLOOR, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

This document specifies the general structure and format of data represented in an EPC™ tag. The specification includes the format of the EPC™ in terms of numbers and bits. The specification also identifies header values, and describes the bit partitions defined for each header.

Published October 1, 2003. Distribution restricted to Sponsors until January 1, 2004.

WHITE PAPER

EPC™ Tag Data Specification

Biography

David L. Brock
Co-Director

Dr. David Brock received Bachelors degrees in theoretical mathematics and mechanical engineering from MIT, and his Masters and Ph.D. degrees from the Department of Mechanical Engineering at MIT with an affiliation to the Artificial Intelligence Lab. He is currently a Principal Research Scientist in the Laboratory for Manufacturing and Productivity and Co-Director of the MIT Auto-ID Center. Dr. Brock is also the Founder of Brock Rogers Surgical, a manufacturer of robotic medical devices. Dr. Brock has worked with a number of organizations including the Artificial Intelligence Laboratory, the Massachusetts Eye and Ear Infirmary, DARPA, Lockheed-Martin, Loral, BBN and Draper Laboratories.

WHITE PAPER

EPC™ Tag Data Specification

Contents

1. INTRODUCTION

The Electronic Product Code™ (EPC™) is a numbering scheme designed to uniquely identify all objects. The EPC™ was created to enumerate all objects and to accommodate current and future numbering methods. Most importantly, the EPC™ was created to enable the electronic connection between physical objects and computer networks; that is to serve as an efficient information reference.

In this document, we describe the encoding specification of the EPC™ in a binary representation; that is a sequence of bits that encodes the Electronic Product Code™ in a well-defined manner.

2. PRINCIPLES OF EPC™ FORMAT

The EPC™ is a unique number designed to identify objects. The EPC™ has a particular form and structure that facilitates uniqueness, number management and information referencing. The binary representation of the EPC™ consists of a particular sequence of partitions and bits that is described in this document.

The EPC™ was intended at the outset to accommodate existing coding standards while maintaining generality, uniqueness, simplicity and efficiency of network addressing. Given these objectives, the EPC™ specification includes a generic **Universal Identifier**, as well as set of **Domain Identifier** that accommodate existing numbering systems.

In the initial version of the EPC™ – EPC™ Version 1.0 – we include the Universal Identifier along with one Domain Identification number that encodes the EAN.UCC Global Trade Item Number (GTIN®).

In the following sections, we will describe the structure and organization of the EPC™ and provide illustrations to show its recommended use.

3. EPC™ GENERAL FORMAT

The general structure of the EPC™ consists of a fixed length header followed by a series of numbers whose structure and function are completely determined by the header value. The EPC™ Specification distinguishes between a generic Universal Identification Number and a series of Domain Identification codes.

For the EPC™ Version 1.0, we specify the Universal Identification number and a single Domain Identification code.

Also note throughout this document we use binary representations of the EPC™ in order to facilitate the embedding of these codes into electronic tags and for the transmission of codes through wireless and wired computer networks. Furthermore, in order to present these binary representations, we will often use hexadecimal notation within the text.

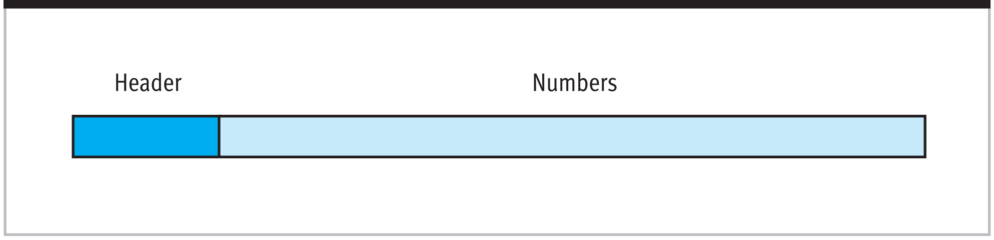
4. EPC™ HEADER

In order to distinguish between types and versions of the EPC™, all EPC™ numbers consist of a fixed length header followed by a series of numbers whose length, organization and structure are determined completely from the header value, as illustrated in Figure 1. In this sense, the header number may be considered **meta-code**; that is it determines the structure and format of the remaining numbers.

Specifically the EPC™ Header (1) determines the structure of the identification code, (2) the number and format of code partitions (including overall length of the code and the length of each partition) and (3) any additional partitions for content or data associated with an EPC™ .

Figure 1: The general form of the EPC™ includes a fixed length header followed by a series of numbers whose form and structure are determined by the header value.

THE GENERAL FOR OF THE EPC™



5. UNIVERSAL AND DOMAIN IDENTIFIERS

The Electronic Product Code™ is intended as a generic, universal identification scheme for objects of all sorts. The EPC™ also serves as a mechanism for efficient referencing of networked information – without consideration for industry or object.

At the same time, however, it is important for the EPC™ to accommodate or "map" into existing numbering standards. In other words, we want the EPC™ to translate directly into current industry codes – such as GTIN®, SSCC, VIN, etc. – without network lookup, external data tables or complex algorithms.

This last objective requires at least some specific structure or data organization within the EPC™ data standard itself to facilitate translation into the various, disparate numbering standards. By directly embedding industries standard codes, we facilitate the rapid adoption of the EPC™ within established industries, and by including a generic, universal identifier we encourage trans-industry cooperation and many new industries that do not have well-defined coding structures.

Therefore, in order to accommodate a generic identification scheme **and** industry specific representations, we propose two "conceptual" EPC™ types – a **Universal Identifier** and a series of **Domain Identifiers**.

By Universal Identifier, we mean simply that – a generic, non-industry specific identification system. This is the identification scheme presented in the original EPC™ proposal.

The Domain Identifiers are simply a series of EPC™ data structures constructed in such a way to make their translation into existing standards as easy and direct as possible. Such industry number standards are well known, and include, for example, the Global Trade Item Number (GTIN®), Serial Shipping Container Code (SSCC), Global Individual Asset Identifier (GIAI), Global Reusable Asset Indicator (GRAI), Global Location Number (GLN), Global Service Relation Number (GSRN), Vehicle Identification Number (VIN), International Standard Book Number (ISBN), National Drug Code (NDC), and many others.

Finally, we say **conceptual** because we do not propose a formal partition or separation of types (Unviversal versus Domain), but simply acknowledge that a particular domain identifier does not necessarily meet the needs as a universal identifier for all industries and all objects.

In summary, we propose the creation of a generic, Universal Identifier and a series of industry specific Domain Identifiers. Specifically, we propose the first Domain Identifier will be the EAN.UCC Global Trade Item Number (GTIN®). In the following sections, we will present the specific structure of both the universal and domain specific Electronic Product Codes™.

6. THE 64-BIT AND 96-BIT ELECTRONIC PRODUCT CODE™

In order to be useful to hardware and software developers, we need to move beyond generalities to specific data representations and numbering structures. Since the Electronic Product Code™ is targeted toward unique object identification and, in particular, embedding on Radio Frequency Identification (RFID) chips, we need to specify exact bit lengths and bit partitions.

Perhaps most important to RFID tag manufacturers is the total data size – or bit length. We propose therefore two EPC™ bit lengths – a 64-bit and 96-bit version. Both versions represent the same code – that is the 64-bit version is a proper subset of the 96-bit version version.

The smaller, 64-bit EPC™ , involves a series of compromises that limit its eventual scope and application. It does, however, lower initial cost and accelerate near-term adoption of the EPC™ infrastructure.

The somewhat larger, 96-bit EPC™ specification significantly expands the range of industry standards that can be accommodated, as well as the number of identification codes. Most importantly, the 96-bit EPC™ greatly simplifies the translation of domain specific codes – such as GTIN® – into their traditional representations.

In the following sections, we will provide specific bit representations of universal and domain identifiers for both the 64-bit and 96-bit version of the Electronic Product Code™.

7. EPC™ SPECIFICATION

7.1. Header

As previously stated, the value of the header identifies the length, type and structure of the Electronic Product Code™. The 64-bit EPC™ uses a 2-bit header and the 96-bit EPC™ an 8-bit header, as shown in Table 1. Given the 64-bit version is a proper subset of the 96-bit, both specifications shared the first two bits.

In this first version of the EPC™, we specify a Universal Identifier and one Domain Identifier, which in this case is the EAN.UCC GTIN® number.

The overall bit length of the EPC™ and the bit length of each partition are established by a given Header value. Four Headers have been defined, as shown in Table 2.

Table 1: The Electronic Product Code™ header specifies the length, type and organization of the code. The 64-bit version uses a 2-bit header and the 96-bit an 8-bit header.

EPC™ TYPE	HEADER LENGTH
96-bit EPC™	8-bits
64-bit EPC™	2-bits

Table 2: The Electronic Product Code™ header specifies the length, type and organization of the code. The 64-bit version 2-bit header and the 96-bit 8-bit header have two values in this specification – one a Universal Identifier and the other a Domain Identifier, which in this case is the EAN.UCC GTIN®.

EPC™ TYPE	HEX VALUE	BINARY VALUE
96-bit Universal Identifier	01	0000 0001
96-bit Domain Identifier (EAN.UCC GTIN®)	10	0001 0000
64-bit Universal Identifier	1	01
64-bit Domain Identifier (EAN.UCC GTIN®)	2	10

7.2. Universal Identifier

The Electronic Product Code™ includes a data type designed for generic, universal, non-industry specific identification. We term this data representation the EPC™ **Universal Identifier**. The Universal Identifier is composed of three partitions in addition to the header – the **Domain**, **Class** and **Instance** numbers, as shown in Table 3. Call it "Manager." Also, why change to "Instance" from "Serial #" given that the latter is a fairly broad usage? Stick with Serial #.

Table 3: The Electronic Product Code™ Universal Identifier includes three partitions in addition to the header – the **Domain, Class** and **Instance numbers**.

BIT ALLOCATIONS PER PARTITION				
	Header	Domain	Class	Instance
96-bit Universal Identifier	8	28	24	36
64-bit Universal Identifier	2	21	17	24

The **Domain** identifies essentially a company, manager or organization; that is an entity responsible for maintaining the numbers in subsequent partitions – Class and Instance. The Domain number is assigned by EPCglobal™ to an entity, and ensures that each Domain number is unique.

The third component is **Class**, and is used by an EPC™ managing entity to identify a class of things. These class numbers, of course, must be unique within every given domain.

Finally, the **Instance** code, or serial number, is unique for every class. In other words, the managing entity is responsible for assigning unique – non-repeating serial numbers for every instance within each object class code.

7.3. Domain Identifiers

The EPC™ specification includes not only a generic, universal identification number, but a series of domain specific identification codes. The first of these **Domain Identifiers** is the EAN.UCC Global Trade Item Number (GTIN®).

As opposed to the Universal Identifier, the EPC™ GTIN® is a slightly more complex construction, reflecting the direct embedding of the EAN.UCC numbering standard into fixed binary codes. Furthermore, the limited address space of the 64-bit EPC™ prohibits a literal embedding of the EAN.UCC GTIN® and has lead to a **mapped** embedding of only **some** of the GTIN® numbers. Because of the somewhat differing structures between 64-bits and 96-bits, we will describe these separately.

7.3.1. 96-bit EPC™ GTIN®

In addition to a Header, the 96-bit EPC™ GTIN® is composed of five partitions: the **Filter Value, Partition, Company Prefix, Item Reference** (including the **Indicator Digit**), and **Serial Number,** as shown in Table 4.

Table 4: The 96-bit EPC™ **Domain Identifier** GTIN® represents a direct embedding of the EAN.UCC numbering standard into the binary code.

	HEADER	FILTER VALUE	PARTITION	COMPANY PREFIX	ITEM REFERENCE	SERIAL NUMBER
96-bit Domain Identifier (EAN.UCC GTIN®)	8	3	3	37–20	7–24	38

After the header, the second number, the **Filter Value** is not part of the EAN.UCC GTIN® specification, but is required for adoption of the EPC™ GTIN® within the consumer products industry, and is used for rapid determination of Filter Value within applications. In other words, the Filter Value is required for fast filtering and pre-selection of basic logistics types, such as items, inner packs, cases and pallets.

The third partition, is actually the **"partition" partition**. This is not a recursive definition, as it might seem, but actually an indication of where the subsequence company prefix and item reference numbers is divided. This organization matches the structure in the EAN.UCC GTIN® in which the Company Prefix (plus the single Package Level Indicator Digit) added to the Item Reference number totals 13 digits, yet the Company Prefix may vary from 6 to 11 digits and the Item Reference from 6 to 1 digit(s) (the Package Level Indicator is always a single digit).

Thus, with the three groupings of numbers – **partition, company prefix** and **item reference** – the 96-bit EPC™ GTIN® literally encodes the EAN.UCC GTIN®. The conversion from the traditional decimal representation of the EAN.UCC GTIN® to the binary representation of EPC™ simply involves a decimal to binary conversion – and visa-versa. Now, the actual division of Company Prefix and Item Reference is shown in Table 5. These divisions essentially mirror the decimal division now used in the EAN.UCC GTIN® .

Table 5: With three groupings of numbers – **partition, company prefix** and **item reference** – the 96-bit EPC™ GTIN® literally encodes the EAN.UCC GTIN®. The partition values in the table mirror the decimal variations in the existing EAN.UCC number.

PARTITION VALUE	COMPANY PREFIX			ITEM REFERENCE AND PACKAGE INDICATOR DIGIT		
	Bits	Decimal Digits	Address Space	Bits	Decimal Digits	Address Space
1	37	11	128 Billion	7	2	128
2	34	10	16 Billion	10	3	1024
3	30	9	1 Billion	14	4	16,384
4	27	8	128 Million	17	5	131,072
5	24	7	16 Million	20	6	1 Million
6	20	6	1 Million	24	7	16 Million

The final partition, **serial number**, is new to the EAN.UCC GTIN®. With the 38-bits allocated, manufacturers can enumerate up to 374,877,906,943 numbers for each product type.

7.3.2. 64-bit EPC™ GTIN®

The 64-bit EPC™ GTIN® involves a series of compromises. The address space afforded by 64-bits is insufficient to literally encode the GTIN®. Literally embedding GTIN® involves a minimum of 48-bits (44-bits for Company Prefix, Item Reference and Package Level Indicator, and 4-bits for Partition) together with 4-bits for Filter Value and 2-bits of header, leaving only 10-bits for serial number – which is insufficient.

The 64-bit EPC™ GTIN® includes **four** partitions in addition to the header – **Filter Value, Company Prefix** and **Item Reference**, as shown in Table 6. As for all 64-bit EPC™ codes, the header occupies 2-bits, and for the 64-bit EPC™ GTIN® this has been assigned the value 0x2, as given in Table 2.

Again **Filter Value** is not part of EAN.UCC GTIN® standard, but was included to facilitate adoption into the consumer products supply chain. The Filter Value partition for 64-bits is identical to Filter Value for the 96-bit code. This occupies the next 3-bits.

The primary compromise of the 64-bit EPC™ GTIN® involves **mapping** the **Company Prefix into** a smaller address space than a **literal** embedding. Specifically, we propose 14-bits for Company Prefix, which allows up to 16,384 manufacturers, as shown in Table 6. For early adoption of EPC™ and for near-term deployment on the smaller 64-bit tag, 16,384 users seem reasonable. Soon thereafter, the 96-bit EPC™ electronic tags will be available and at low-cost allow a much wider adoption.

Table 6: The 64-bit EPC™ Domain Identifier GTIN® represents a mapped embedding of the EAN.UCC numbering standard into the binary code.

	HEADER	FILTER VALUE	COMPANY PREFIX	ITEM REFERENCE	SERIAL NUMBER
64-bit Domain Identifier (EAN.UCC GTIN®)	2	3	14	20	25

The next partition, **Item Reference**, is a literal embedding of the EAN.UCC GTIN® Item Reference into the 64-bit EPC™ GTIN®. For the early adaptors, initial trials and larger manufacturers, the 20-bits allocated for Item Reference will be sufficient to literally encode the decimal representation of the EAN.UCC GTIN® Item Reference.

As with the 96-bit EPC™ GTIN®, the Item Reference number includes the single Indicator digit, which is specified in the EAN.UCC GTIN® EAN/UCC-14.

Finally, 64-bit EPC™ GTIN® includes 25-bits for serial number, which allows up to 33,554,431 individual numbers for each product class. Again for **near-term** adoption on the smaller 64-bit tag, this should be sufficient.

8. REFERENCES

This document draws from the previous work at the Auto-ID Center, and we recognize the contribution of the following individuals: David Brock (MIT), Joe Foley (MIT), Sunny Siu (MIT), Sanjay Sarma (MIT), Dan Engels (MIT). In addition, we recognize the contribution from Steve Rehling (P&G) on EPC to GTIN® mapping.

The following papers capture the contributions of these individuals:

1. **D. Engels, J. Foley, J. Waldrop, S. Sarma & D. Brock, "The Networked Physical World: An Automated Identification Architecture".**
   Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD01), 76–77, 2001.

2. **D. Brock, "The Electronic Product Code (EPC), A Naming Scheme for Physical Objects".**
   2001.

3. **D. Brock, "The Compact Electronic Product Code; A 64-bit Representation of the Electronic Product Code".**
   2001, http://www.autoidcenter.org/publishedresearch/MIT-AUTOID-WH-002.pdf