# Introduction

RAIN RFID enables digital life for everyday items by storing data on a wireless RAIN RFID tag. The structure and format of the data stored on a RAIN RFID tag is called the numbering system, and it is often part of a broader data standard.

This document describes best practices for choosing and implementing a numbering system that ensures the success of your RAIN RFID deployment. The intent is to provide a high-level overview of current numbering systems in use, the importance of encoding the Protocol Control (PC) Word, and practical examples of how to use the new RAIN Alliance ISO Numbering System.

# Why Use Standardized Numbering Systems?

Successful RAIN RFID deployments leverage numbering systems that are part of data standards managed by organizations such as GS1 and ISO because these numbering systems:

* Eliminate tag clutter: Allows easy filtering of “my tags” from other stray reads, e.g. quickly filter package level tags from tags on items inside the same package.
* Ensure uniqueness: Following a numbering system can prevent different organization from unknowingly encoding duplicate tags.
* Facilitate resolution: The use of numbering systems allows readers to quickly and efficiently resolve tag data to gain additional item information when desired.
* Support interoperability: Uniformity of data structures supports supply chain partnerships
* Efficiently use memory space: RAIN RFID tags have limited memory, so data must be encoded efficiently.

As the prevalence and size of RAIN RFID deployments grow and tag read range increases, you are more likely to encounter tags in your everyday life or— if you have a RAIN RFID deployment —in your read zone. Combined, these factors underscore the importance of the broad adoption of standardized numbering systems to avoid interference with neighboring RAIN RFID deployments.

# How to Choose a Standardized Numbering System

Choosing the right numbering system for RAIN RFID tag data is one of the most important aspects of any deployment. The RAIN Alliance recommends this 3 step process for selecting a tag numbering system:

1. **If there’s an applicable GS1 standard for your application space – follow it.**
   * GS1 is active in many industries adopting RAIN RFID—including retail, healthcare, logistics, and food.
   * See here for more on GS1 tag data standards: <https://www.gs1.org/standards/tds>
2. **If GS1 is not active in your application space, look for ISO-based industry specific organizations for tag data standards**
   * Examples of organizations with ISO numbering systems include IATA for aviation, ATA for aerospace manufacturers, VDA for German automotive, AIAG for North American automotive, MIL-STD 129 for U.S. Department of Defense, and ISO/IEC 20248 for secure digital signatures and vehicle identification.
   * For a complete listing of ISO numbering system options see: <https://www.aimglobal.org/uploads/1/2/4/5/124501539/iso-iec_15961-2_data_constructs_register_2021-09-22.pdf>
3. **For \*anything else\*, including asset tracking, closed-loop logistics, or any use case that isn’t covered by options #1 and #2, use the RAIN Alliance ISO numbering system.**
   * This numbering system allows for maximum flexibility for diverse RAIN RFID use cases, while ensuring compatibility with existing GS1 encoding standards.
   * See here for more info: <https://rainrfid.org/wp-content/uploads/2022/05/RAIN-Alliance-Tag-Encoding-Guideline.pdf>

# Registering for a RAIN Alliance Company Identification Number

The first step in implementing the RAIN ISO numbering system is registering for a Company Identification Number (CIN) with the RAIN Alliance.

4 lengths of CINs are available, to cover deployments of a few hundred tags to hundreds of millions.

|  |  |  |
| --- | --- | --- |
| **CIN Length** | **Unique numbers available with 96 bit UII/EPC memory** | **Recommended deployment size** |
| 2 digits / 8 bits | ~3 × 1026 | ≥ 100,000,000 tags/year |
| 4 digits / 16 bits | ~1 × 1024 | ≥ 1,000,000 tags/year |
| 6 digits / 24 bits | ~5 × 1021 | ≥ 10,000 tags/year |
| 8 digits / 32 bits | ~2 × 1019 | ≥ 100 tags/year |

Select the CIN length appropriate for your deployment size. More information on the CIN registration process, fees, and currently issued CINs is available from the RAIN Alliance here - <https://rainrfid.org/cin/>

The RAIN Alliance will provide the issued CIN in a few different formats: hexadecimal, decimal, and Extensible Bit Vector (EBV-8) hexadecimal. Most encoding software works in hexadecimal, while decimal is the ISO unit. The EBV-8 format allows for variable length data up to 32 bits, and ensures uniqueness of tag data for all CIN lengths. If you’re unfamiliar with EBV-8 and the conversion between decimal and hexadecimal, there are additional details at the end of this document.

# The PC Word and AFI bits

All standardized numbering systems require encoding the Protocol Control (PC) Word. It is defined by the air interface standard to provide meta-information about the data stored on the tag, including the length of the encoded data and what numbering system is used. The PC Word is stored in Memory Bank 01, typically referred to as “EPC/UII memory” starting in word 1. Part of its role is to provide a “preview” of the type and amount of data stored elsewhere in the tag memory. The air interface standard is available here: https://www.gs1.org/standards/rfid/uhf-air-interface-protocol

When reading a tag as part of an “Inventory Round”, the tag provides the PC Word and EPC/UII data in its response to the reader. Some readers and reader software do not show the PC Word by default and may need to be configured to do so.

Note GS1 and ISO standards use different terminology – EPC and UII refer to the same memory location, EPC for GS1 and UII for ISO.

Timeline

Description automatically generated with low confidence

Figure 1. EPC/UII memory bank contents, including the PC Word.

There are 5 information elements encoded in the PC Word:

* **Length:** 5 bits that define the amount of data stored in the EPC/UII memory bank. For example, a tag that has 8 words (128 bits) of available EPC memory, may only be encoded with 6 words (96 bits) of information. The Length bits tell the tag how much of the EPC memory to respond to an inventory command with.
* **User Memory Indicator (UMI):** 1 bit yes/no indicator for the presence of additional User Memory. This bit is determined by the tag IC itself.
* **Extended Protocol Control (XPC) Indicator**: 1 bit yes/no indicator for the presence of advanced features. This bit is determined by the tag IC itself and is set to 02 for most RAIN tags.
* **Toggle (T) bit:** 1 bit yes/no indicator for the type of numbering system. Set as 02 for GS1 numbering systems by default. Should be changed to 12 for ISO numbering systems.
* **Attribute bits / Application Family Identifier**:
  + **For GS1 encoded tags with T=02**, current GS1 standards refer to this data field as attribute bits “reserved for future use” and should be set to zero.
  + **For non-GS1 or ISO tags with T=12**, this data field is known as the Application Family Identifier (AFI) – it serves to define the numbering system, application or industry of the tag data. The AFI for the RAIN Alliance ISO numbering system is 0xAE (See here for others [ISO/IEC 15961-2 Data Constructs Register](https://www.aimglobal.org/uploads/1/2/4/5/124501539/iso-iec_15961-2_data_constructs_register_2021-09-22.pdf) )

When using the RAIN ISO numbering system, it’s critical to set the Toggle bit (T) to 12 and the AFI bits to 0xAE. If you’re not familiar with how to calculate the PC Word, there’s a table of common PC word values for reference at the end of this document.

# Practical Examples using the RAIN Alliance ISO Numbering System

Use of access or kill passwords, user memory, locking, permalocking, and other advanced RAIN RFID features are not covered in this document. In most cases, these additional tag features do not impact selection of a numbering system.

**Example 1:**

***I have an existing barcode-based asset tracking system that uses a prefix of 4 letters, follow by 6 numbers. These assets never leave my organization, there are 10,000 of them currently and I add ~100 more each month. Example asset ID numbers are: ABYZ123456, CDWX789012***

For a deployment of this size, choose a 6-digit / 24-bit RAIN CIN length. We’ll assume an RFID tag with 128 bits EPC/UII memory, no User Memory, and no XPC features.

First, select the PC Word value for: Length= 8 words (128 bits), UMI=02, XPC=02, T=12 AFI=0xAE. From the table, that would be 0x41AE.

We’ll assume the assigned CIN from RAIN is 123456, which in hexadecimal is 0x87C440.

With 128 bits EPC/UII memory available, and a RAIN CIN of 24 bits, that leaves 104 bits for my asset data. But, first we have to decide how to store the alphanumeric data in hexadecimal or binary.

One option is to use 8-bit ASCII encoding (<https://en.wikipedia.org/wiki/ASCII#8-bit_codes>). This approach assigns each number, uppercase and lowercase letter, and some punctuation characters an 8 bit or 2 Hexadecimal character code. Many RAIN RFID readers, printer/encoders, and software tools can convert ASCII characters to hexadecimal for either encoding or reading tag data. There are also many on-line tools available for quick data conversion.

With 10 alphanumeric characters in my asset IDs, converting them to hexadecimal will result in 80 bits or 20 hex characters. But, I have 104 bits of available data, which is enough space for 13 characters. In this example, we’ll just pad the original asset IDs with some extra zeros. Another option would be to use a shorter length value in the PC word.

|  |  |
| --- | --- |
| ASCII Asset ID | Hexadecimal Asset ID (80 bits) |
| ABYZ123456 | 0x4142595A313233343536 |
| CDWX789012 | 0x43445758373839303132 |

|  |  |
| --- | --- |
| ASCII Asset ID with padded zeros | Hexadecimal Asset ID (104 bits) |
| ABYZ123456000 | 0x4142595A313233343536303030 |
| CDWX789012000 | 0x 43445758373839303132303030 |

Now I can construct the complete EPC/UII:

|  |  |  |  |
| --- | --- | --- | --- |
| **PC Word** | **RAIN CIN Header** | **Asset ID Hexadecimal** | **Original Asset ID** |
| 0x41AE | 0x01E240 | 0x4142595A313233343536303030 | ABYZ123456 |
| 0x41AE | 0x01E240 | 0x43445758373839303132303030 | CDWX789012 |

**Example 2.**

***I run a record keeping system for a large law firm. We need to track more than 1 million documents per year. Our current document IDs are 16 characters long and alphanumeric. Example Document ID numbers are: 1234ABCD-4567EFGH and WXZY1234-QRST5678***

For a deployment of this size, choose a 4-digit / 16-bit RAIN CIN length. Example 1 used an 8-bit ASCII encoding scheme, but with 17 characters (including the dash) that would require 152 bits -- 136 bits for the data and 16 bits for the CIN. While there are tag ICs available with that amount of EPC/UII memory, this example will show how to use Base36 encoding ( <https://en.wikipedia.org/wiki/Base36> ) and a more commonly available 128 bit size EPC/UII.

Like example 1, assume an RFID tag with 128 bits EPC/UII memory, no User Memory, and no XPC features will be used. Also assume the assigned 4-digit CIN from RAIN is 1234, which in hexadecimal is 0x8952. Before deciding on the PC Word, we need to determine how to encode the document IDs.

Base36 uses the numbers 0-9 and capital Latin letters A-F. It requires <6 bits per character, and for RFID encoding in hexadecimal it works best for data lengths that are a multiple of 8. In this example we’ll encode our 16-digit document IDs in 2 groups of 8. Conversion between hexadecimal and Base36 is not as widely supported as ASCII by RFID software, but it’s a good option encoding scheme for alphanumeric data in RAIN RFID tags.

It's straight forward to convert between Base36 using an [on-line tool](https://www.multiutil.com/base36-number-converter/), excel, or programming script. Each group of 8 Base36 characters are converted to 12 hexadecimal characters, which takes 96 bits total. With our 16-bit RAIN CIN, that makes the total data length 112 bits. Select the PC Word value for: Length= 7 words (112 bits), UMI=02, XPC=02, T=12 AFI=0xAE. From the table, that would be 0x39AE.

|  |  |  |  |
| --- | --- | --- | --- |
| **PC Word** | **RAIN CIN Header** | **Asset ID in Hexadecimal** | **Document ID in Base36** |
| 0x39AE | 0x04D2 | 0x134D9A27ED 4B9A91D4C1 | 1234ABCD 4567EFGH |
| 0x39AE | 0x04D2 | 0x25916F59ED0 1E8798E0BA4 | WXZY1234 QRST5678 |

When reading the tags, make sure to use the same process in reverse – converting the hexadecimal Asset IDs back into Base36 in the same groups of 12 characters.

**Example 3.**

***I want to track company IT assets and both a 6-digit alpha numeric asset ID and a 48-bit device MAC address on a 128 bit RAIN RFID tag. I expect to use about 500 tags per year. Examples asset IDs are ABC123 and 789XYZ. Example MAC addresses are 00-14-22-01-2C-45 and 00-40-96-A4-F1-34.***

For a deployment this size, choose an 8 digit / 32-bit CIN length. The 6-digit asset IDs can be encoded using 8-bit ASCII with 48 bits, followed by the 486bit MAC address which is already represented in hexadecimal.

Assume we’re using a tag with 128 bits of EPC memory, 32 bits of User memory, and no XPC features. Select the PC Word value for: Length= 8 words (128 bits), UMI=12, XPC=02, T=12 AFI=0xAE. From the table, that would be 0x45AE.

We’ll assume the assigned CIN from RAIN is 12345678, which in hexadecimal is 0x85F1C24E.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PC Word** | **RAIN CIN Header** | **Asset ID + MAC in Hexadecimal** | **Asset ID** | **MAC Address** |
| 0x45AE | 0xBC614E | 0x414243313233 **001422012C45** | ABC123 | **00-14-22-01-2C-45** |
| 0x45AE | 0xBC614E | 0x37383958595A 004096A4F134 | 789XYZ | 00-40-96-A4-F1-34 |

**Example 4.**

***I run a logistics business with parcel volume exceeding 100 million packages per year. How can I use the RAIN ISO numbering system to identify packages within my network?***

For a deployment this size, choose a 2 digit / 8-bit CIN length. With a commonly available tag with 128 bits of EPC/UII memory, that would leave 120 bits to encode a package IDs. If used efficiently 120 bits can store 2120 or 1.3×1036 unique numbers – which is an unfathomably large number!

With the RAIN ISO number system, these 120 bits can be encoded using any method desired. One commonly used approach is to divide the space into sections that represent different attributes. For example, you might choose to segment the EPC/UII data like this:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **RAIN CIN** | **Service Level** | **Time Stamp** | **Origin Code** | **Destination Code** | **Package Identifier** |
| Encoded Length | 8 bits | 8 bits | 24 bits | 20 bits | 20 bits | 48 bits |
| Numbers available | 90 | 256 | ~16.7 million | ~1 million | ~1 million | ~280 trillion |
| Purpose | Identifies tag owner | Freight, Ground, Air, Overnight, Priority, etc. | Identify every minute since 1/1/2022 for ~32 years | Identify every US ZIP code, with room to grow | | Uniquely identify 280 trillion packages |

Again assume we’re using a tag with 128 bits of EPC memory, 32 bits of User memory, and no XPC features. From the table select the PC Word value for Length= 8 words (128 bits), UMI=12, XPC=02, T=12 and AFI=0xAE, which is 0x45AE. We’ll assume the assigned CIN from RAIN is 12, which in EBV-8 hexadecimal is 0x0C.

For a package shipped at 8am on October 18, 2022 from Chicago ZIP code 60610 to Denver ZIP code 80202 with service level of 0xA5, example tag data might look like this:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PC Word** | **RAIN CIN Header** | **Service Level** | **Time Stamp** | **Origin Code** | **Destination Code** | **Package Identifier** |
| 0x45AE | 0x0C | 0xA5 | 0x066120 | 0x0ECC2 | 0x1394A | 0x123ABC456DEF |

# Common Questions

***I encoded ASCII data to my tag, so why is my reader showing me something different?***

Most RAIN RFID readers and reader software report tag data in Hexadecimal by default. If you’ve encoded data using the RAIN ISO numbering system in ASCII, you’ll need to convert the data after the CIN header from hexadecimal back into ASCII.

***Why isn’t my reader showing me the PC word?***

Some RAIN RFID readers and reader software don’t show the PC Word by default. Consult with your reader or software provider to make sure they can support filtering tags by the PC Word.

***Why can’t I encode a shipping address in an RFID tag?***

RAIN RFID tags have limited amount of memory, the most widely used only have ~128 bits of encodable memory. While that’s enough space to uniquely identify a mind-bogglingly large number of items, it’s only enough for 16 ASCII characters.

***How do I convert between Decimal, Hexadecimal and Binary?***

There are many tools available, including Microsoft Excel conversion functions, the Windows calculator (in programmer mode), or numerous on-line tools.

# Glossary of Terms and Acronyms:

* + Air Interface: The complete communication link between a reader and a tag including the physical layer, collision-arbitration algorithm, command and response structure, and data-coding methodology.
  + AFI: Application Family Identifier, bits 18-1F hexadecimal in the EPC memory bank for ISO numbering systems.
  + ASCII: American Standard Code for Information Interchange, is a charter encoding standard for electronic communication. Includes uppercase letters (A-Z), numbers (0-9), and symbols (such as “!” and “\”)
  + Attribute bits: bits 18-1F hexadecimal in the EPC memory bank for GS1 numbering systems.
  + Binary: a base-2 numbering notation, sometimes denoted by a subscript “2” after the data value (for example, “12” means binary 1).
  + Bit: the smallest unit of data that a tag chip can process and store. A bit is always in one of two physical states, similar to an on/off light switch. The state is represented by a single binary value, usually a 02 or 12.
  + Byte: 8 bits
  + CIN: Company Identification Number, a unique value assigned by an ISO “Issuing Agency” to identify a company or entity. The RAIN Alliance is an ISO-authorized Issuing Agency and can issue RAIN CINs to organizations encoding tags with the RAIN ISO numbering system.
  + CRC: cyclic-redundancy check, a data field used by a tag (or reader) to ensure the validity of data communicated by a reader (or tag).
  + EBV-8: Extensible Bit Vector format – 8 bits.
  + EPC: Electronic Product Code. GS1 terminology for the unique product identifier stored in Memory Bank 01. Often used to refer to the entire memory bank (see “EPC memory bank” below). ISO terminology equivalent is “UII” (see below).
  + EPC memory bank: One of four memory banks on a RAIN RFID tag that stores business information such as the physical item to which the tag is attached. Also known as EPC/UII memory, Memory Bank 01, and MB-01, this memory bank contains the CRC, PC, and EPC/UII (and XPC if supported by the tag chip).
  + GS1: a neutral, global collaboration platform that brings industry leaders, government, regulators, academia, and associations together to develop standards-based solutions to address the challenges of data exchange.
  + Hexadecimal: a base-16 numbering notation, sometimes denoted by “0x” preceding the value (for example, “0x1A” means hexadecimal 1A).
  + IC: integrated circuit, also known as a tag chip.
  + IEC: International Electrotechnical Commission, an organization that sets international electrical and electronics standards.
  + Interrogator: a RAIN RFID reader
  + Inventory Round: The period initiated by a Query command and terminated by either a subsequent Query command (which also starts a new inventory round), a Select command, or a Challenge command.
  + ISO: International Organization for Standardization. An independent, non-governmental international organization with a membership of 167 national standards bodies.
  + PC Word: Protocol Control Word, word 1 (bits 10-1F hexadecimal) in the EPC/UII memory bank.
  + Protocol: a physical layer and a Tag-identification layer specification. In this document, “protocol” specifically refers to the GS1 EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard or ISO/IEC 18000-63 Information technology - Radio frequency identification for item management - Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C. Also commonly known as “Gen2.”
  + Query: a Gen2 command that initiates an Inventory Round.
  + Sensitivity: reader sensitivity is the weakest tag signal that a RAIN RFID reader is able to detect. Tag sensitivity is the weakest reader signal that is capable of turning on a RAIN RFID tag.
  + Tag Data: the business, control, and tag manufacturer information stored in a tag chip.
  + Toggle Bit: Bit 17 hexadecimal in the EPC/UII memory bank, also known as the T bit
  + UII: Unique Item Identifier, item information beginning at bit 20 hexadecimal in the EPC/UII memory bank for ISO numbering systems. UII is the ISO equivalent of the EPC in a GS1 numbering system.
  + UMI: User Memory Indicator, bit 15 hexadecimal in the EPC/UII memory bank.
  + Word: 16 bits
  + XI: Extended Protocol Control (XPC) Indicator.

# Useful links and references:

* GS1 RFID
  + <https://www.gs1.org/standards/rfid>
* List of 8 bit AFI Assignments
  + <https://www.aimglobal.org/uploads/1/2/4/5/124501539/iso-iec_15961-2_data_constructs_register_2021-09-22.pdf>
* RAIN ISO+CIN Standard:
  + [https://rainrfid.org/cin/](https://rainrfid.org/technology/rain-alliance-iso-numbering-system/)
  + <https://rainrfid.org/technology/rain-alliance-iso-numbering-system>
* RAIN Relevant Standards List
  + <https://rainrfid.org/wp-content/uploads/2022/05/RAIN-RFID_TWG_RAIN_RFID_Relevant_Standards_FINAL_20220503-v1.3.pdf>

# Extensible Bit Vector EBV-8 Encoding

EBV-8 encoding format EBV (Extensible Bit Vector) is a data structure with an extensible range. An EBV is an array of blocks with each block containing a single extension bit followed by a specific number of data bits. For EBV-8, there are 8 bits in one block and each block contains an extension bit followed by 7 data bits. The data value represented by an EBV-8 is simply the bit string formed by the data bits as read from left-to-right, ignoring the extension bits. Because each block has 7 available data bits, an EBV-8 can represent numeric values between 0 and 127 with a single block. To represent the value 128, set the extension bit to 1 in the first block, and append a second block to the EBV-8. In this manner, an EBV-8 can represent arbitrarily large data values. RAIN CIN values are in the range 0 to 99,999,999 so the corresponding EBV-8 encoding will require from 1 to 4 bytes of memory.

Table

Description automatically generated

# Table of common PC Word Values using the RAIN AFI

The following table shows valid PC word values in binary and hexadecimal for Toggle Bit = 12, XPC = 02, AFI = 0xAE, and UMI = 02 or 12.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **UMI = 0** | | **UMI =1** | |
| **EPC/UII Length (bits)** | **PC Word (Binary)** | **PC Word (Hexadecimal)** | **PC Word (Binary)** | **PC Word (Hexadecimal)** |
| 0 | 00000001 | 01AE | 00000101 | 05AE |
| 16 | 00001001 | 09AE | 00001101 | 0DAE |
| 32 | 00010001 | 11AE | 00010101 | 15AE |
| 48 | 00011001 | 19AE | 00011101 | 1DAE |
| 64 | 00100001 | 21AE | 00100101 | 25AE |
| 80 | 00101001 | 29AE | 00101101 | 2DAE |
| 96 | 00110001 | 31AE | 00110101 | 35AE |
| 112 | 00111001 | 39AE | 00111101 | 3DAE |
| 128 | 01000001 | 41AE | 01000101 | 45AE |
| 144 | 01001001 | 49AE | 01001101 | 4DAE |
| 160 | 01010001 | 51AE | 01010101 | 55AE |
| 176 | 01011001 | 59AE | 01011101 | 5DAE |
| 192 | 01100001 | 61AE | 01100101 | 65AE |
| 208 | 01101001 | 69AE | 01101101 | 6DAE |
| 224 | 01110001 | 71AE | 01110101 | 75AE |
| 240 | 01111001 | 79AE | 01111101 | 7DAE |
| 256 | 10000001 | 81AE | 10000101 | 85AE |
| 272 | 10001001 | 89AE | 10001101 | 8DAE |
| 288 | 10010001 | 91AE | 10010101 | 95AE |
| 304 | 10011001 | 99AE | 10011101 | 9DAE |
| 320 | 10100001 | A1AE | 10100101 | A5AE |
| 336 | 10101001 | A9AE | 10101101 | ADAE |
| 352 | 10110001 | B1AE | 10110101 | B5AE |
| 368 | 10111001 | B9AE | 10111101 | BDAE |
| 384 | 11000001 | C1AE | 11000101 | C5AE |
| 400 | 11001001 | C9AE | 11001101 | CDAE |
| 416 | 11010001 | D1AE | 11010101 | D5AE |
| 432 | 11011001 | D9AE | 11011101 | DDAE |
| 448 | 11100001 | E1AE | 11100101 | E5AE |
| 464 | 11101001 | E9AE | 11101101 | EDAE |
| 480 | 11110001 | F1AE | 11110101 | F5AE |
| 496 | 11111001 | F9AE | 11111101 | FDAE |