

RAIN URI Identifier

(Issued: 2025-12-15)

1 Method

URI, especially HTTPS, item identifiers are increasingly encoded in barcodes and NFC tags. Mobile smart device operating systems support the reading of such URI data constructs using the device imager or NFC reader without the need for an additional App. The RAIN URI Identifier ensures that a RAIN item identifier, in future, can be read and resolved without additional applications by device operating systems to access the cloud-based item information.

The RAIN URI Identifier can be used in two modes:

1. **URI Identifier mode:** As a unique item identifier (UII) stored in MB01 and optionally extended in MB11 (UserMem).

When the UII is not extended to MB11, then other DSFID specific data may be encoded in MB11, in which case the DSFID shall be the first byte of MB11.

2. **URI Add-on mode:** As an "add-on" to an existing UII with the URI stored in MB11 (User Memory), subject to the rules of the identifier as indicated by the PC-Word (T-bit and AFI). EXAMPLE For an ISO/IEC 17360 UII, the URI stored in MB11 is prepended to the UII according to the rules of this document.

The RAIN URI Identifier read scenarios are discussed in Annex E.

2 Terms and conventions

2.1 Terms

Term	Description	Specification
0xH...H H..H _{hex}	A hex presented binary string, with 'H' representing a '0' to '9' and 'A' to 'F'.	ISO/IEC 9899
AFI	Application Family Identifier, which is part of the PC-Word.	ISO/IEC 19762
BaseURI	The part of the URI before the query, i.e. the Scheme, Authority/host and optional Path.	Term based on RFC 3987
DSFID	Data Storage Format Identifier.	ISO/IEC 15961
Gen2V2 Gen2V3	Versions of the GS1 EPC® Radio-Frequency Identity Generation-2 UHF RFID Standard.	Gen2V3 is fully incorporated in ISO/IEC 18000-63
Inventory	The process of reading the UII.	ISO/IEC 18000-63
MBxx	Memory Banks 00, 01, 10 and 11 of a RAIN tag.	ISO/IEC 18000-63
PC PC-Word	Protocol Control word (16 bits) which is transmitted leading the UII. Provides metadata about a tag's encoding.	ISO/IEC 18000-63

UII	Unique Item Identifier. The ISO equivalent to GS1's term "EPC".	ISO/IEC 19762
UserMem User Memory	MB11 of a RAIN tag.	ISO/IEC 18000-63
UTF-8	Unicode Transformation Format, 8-bit	ISO/IEC 10646, RFC 3629
XPC XPC-word XPC_W1/W2	The optional Extended Protocol words, XPC_W1 and XPC_W2, which are transmitted, when present, following the PC-Word but before the UII.	ISO/IEC 18000-63
X...X ₂	A bit presented binary string with 'X' representing a '0' or '1'. Separators are used for reliability.	-
<u>XXX</u>	The shading indicates a control character.	-
<u>XX...X</u>	Underlined text in URIs indicates that the data is encoded on the tag.	-

2.2 Binary data presentation

Binary shall be presented in lowercase-hex (0-1 and a-f).

The hex representations shall be leading-zero-bit padded to an 8-bit (byte) boundary.

NB Care must be taken when encoding the tag where the encoding does not end on a 16-bit word boundary. RAIN encodes and reads tags in multiples of 16-bit words. This means that such encodings are zero padded to a 16-bit boundary of the full encoding.

3 Encoding identification

The T-bit in the PC-Word shall be set to 1₂.

The AFI shall be set to 0xBD.

When encoded in User Memory (MB11), then the DSFID 35 (0x24) shall be used to

1. identify a URI extension when the AFI is set to 0xBD, otherwise
2. it indicates that MB11 (User Memory) contains a URI to be prepended to the identifier as contained in MB01 (UII/EPC) according to the rules of that identifier, unless specified in this document.

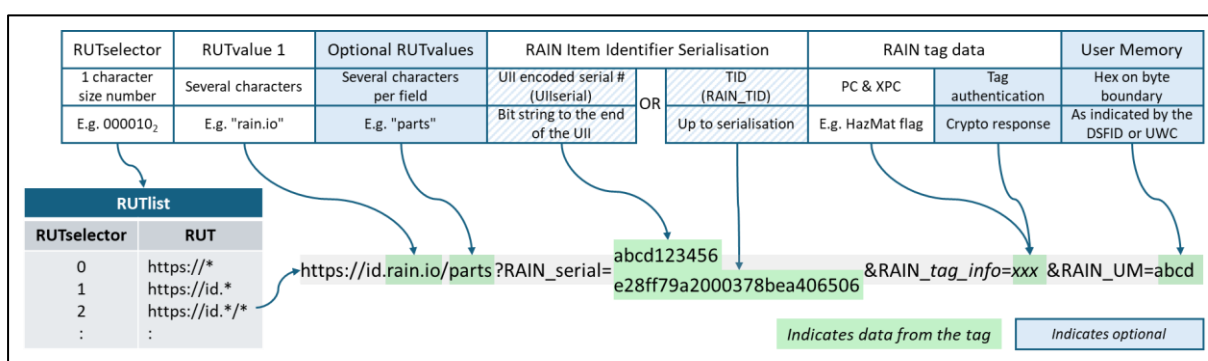
4 The RAIN URI Identifier syntax

4.1 The encoding

The RAIN URI Identifier is compliant with RFC 1738 and RC 3987. To reduce the size of the encoding the following apply:

1. The RAIN URI Identifier uses a RAIN URI template (RUT) with RUTvalue placeholders, see 0. The RUTvalues are encoded in the RAIN tag.

- There may be zero or more RUTvalues encoded in MB01 (UII) and optionally to MB11 (UserMem). The RUTvalues shall be mapped to the RUTvalue placeholders in sequence.
 - RUTvalues without a placeholder shall be ignored.
- 2. RUTvalue placeholders without a RUTvalue shall use the RUTvalue 'null'.
- Encoding: 6-bit characters for MB01 (UII), see Annex A, and UTF-8 for MB11 (UserMem).
- 3. The RAIN URI Identifier uses query fields to report tag data like sensor data and tag authentication.



4.2 The URI construct

The RAIN URI Identifier uses a RAIN URI Template (RUT) of which the list of RUTs is managed and published by the RAIN Alliance, see 0.

NOTE 1 The URI specified format according to RFC 3987 is:

'scheme:[//[user:password@]host[:port]]/[/]path[?query][#fragment]'

The 'user:password', 'port' and 'fragment' is not used in this specification.

The RAIN URI Identifier is constructed with the following components:

- A RAIN URI Template (RUT), of which the list of registered RUTs is published and managed by the RAIN Alliance, see 0. The RUT to be used is indicated by the encoded RUTselector.
- The encoded RUTvalues (data fields of 6-bit characters) are used to populate the RUT.
- Query keys and values are used to report tag specific information, see clause 0.

The RAIN URI is:

<RUTselector> → <RUT> + <RUTvalues> + <RAIN query keys and values>

The RUTlist have the following records:

[<RUTselector>, <RUT>]

Where the RUTselector is a 6-bit binary number which is the first "character" of the encoding. The RUT contains placeholders indicated by an asterisk * to be replaced by the encoded RUTvalues, see Annex B.



EXAMPLE 1 The RUTlist contains the record: [1, 'https://id.*.io'].

The tag Ull is encoded with: '000001₂rainSTE_{6-bit text}'. The tag TID is encoded with 'E28FF79A2000378BEA406506_{hex}'.

Ull binary encoding: RUTselector → 000001₂, r → 010010₂, a → 000001₂, i → 001001₂, n → 001110₂, STE → 111100₂

Ull binary: ||0000 0101|0010 0000|0100 1001|0011 1011||1100 0000|0000 0000||₂

Zero bit padding is added to the next 16-bit word boundary.

Ull in hex: '0520 493B C000_{hex}'

The URI reported is 'https://id.rain.io?RAIN_serial=e28ff79a2000378bea406506'.

The RAIN URI Identifier syntax (after tag data decoding) is as follows:

<BaseURI>?<query>

- BaseURI is the URI scheme, host and path as constructed using the RUT populated with the encoded RUTvalues.
- The BaseURI shall not use capital letters.
- RAIN_<query> is a series of key-value pairs, forming the query component of an URI according to RFC 3986, see clause 0.
- The URI shall use the %-encoding as specified in RFC 3986 to ensure the URI is compliant. The use of characters specified to be %-encoding in the URI should be avoided.

NOTE The combination BaseURI with TID as the serialisation provides a globally unique identifier for the item. This approach provides for web-based Digital Twin functionality in limited memory tag chips.

EXAMPLE 2 In the following examples build on example 1.

The Ull is encoded with '000001₂rainSTE_{6-bit text}' and the TID is 'E28FF79A2000378BEA406506_{hex}'. The RUT for RUTselector 1 is 'https://id.*.io'. In the first two examples below, XPC values was returned with the PC and Ull during inventory. In the third example the UMI/RUM-bit is set in the PC. The reader therefore read and reported MB11 (UserMem). The reader added the optional read-timestamp in the second example.

'https://id.rain.io?RAIN_TID=e28ff79a2000378bea406506&RAIN_hazmat=true'

'https://id.rain.io?RAIN_TID=e28ff79a2000378bea406506&RAIN_sensoralarm=true&RAIN_readtimestamp=20250406T142355'

'https://id.rain.io?RAIN_TID=e28ff79a2000378bea406506&RAIN_UM=8312ab09d8cd'

5 URI Identifier mode

5.1 The URI

In this mode the RAIN URI Identifier is the primary identifier of the item. It is encoded as the UII in MB01 and received during inventory of the tag. MB11 (UserMem) may contain an extension of the encoding indicated by the DSFID 0x24, or other DSFID indicated data constructs.

The RAIN URI Identifier presentation is constructed by populating the selected RAIN URI Template (RUT), see Annex B, with the encoded RUTvalues.

The RAIN_serial shall be included in the URI either as a query or mapped in the path as specified by the selected RUT.

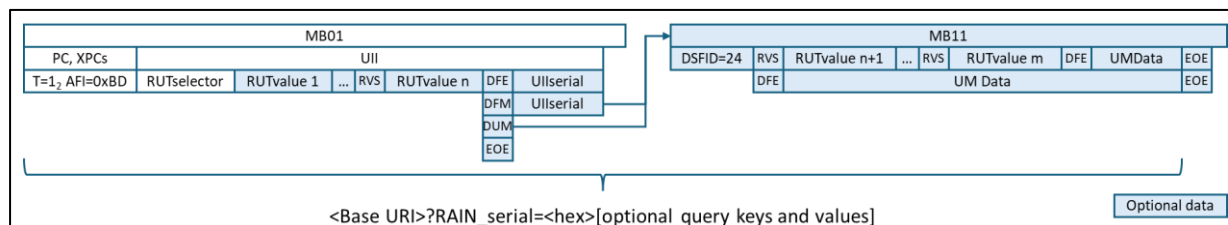
The URI is completed by performing the read scenario indicated tag access and adding the results as URI query key-value pairs.

The RAIN_serial shall be reported in the URI, either within the path or the query of the URI as indicated by the selected RUT, see 0. The following applies:

- RAIN_serial is a binary number.
NOTE Numbers conventionally pads with leading zeros and strings pads with trailing zeros or spaces. This maintains number and string values independent of its length.
- The RAIN_serial shall be reported using lowercase-hex on a byte-boundary, i.e. leading zero-bit padding shall be used.
- The RAIN_serial may be mapped to the path using a RUT.
- The RAIN_serial value may be encoded in MB01 as part of the UII (UIIserial), if not, TID up to and including the serialisation shall be used, i.e. the TID tag declarations are not included in the value.

5.2 The encoding

The encoding is depicted in the following diagram.



The RUTselector is a 6-bit number. It is used to select the desired RUT from the RUTlist.

The RUTvalue shall use character encoding:

- MB01: 6-bit according to the character set specified in Annex A. The RUTselector is a 6-bit value.
- MB11: UTF-8 with the control character mapping set specified in Annex A.



NOTE 1 The 6-bit character set (including the lowercase letter limitation) accommodates 96- and 128-bit commodity tags. A single placeholder RUT can have a RUTvalue of 14 characters long in a 96-bit Ull, 19 characters in a 128-bit Ull, with the maximum length 74 characters.

Ullserial is a binary number. Ullserial is optional; when present, Ullserial shall:

- Be encoded as a bit string in MB01 following SFE or SFU control characters.
- The Ullserial bit encoding ends with the Ull encoding as received during Inventory.
- When encoded the Ullserial shall be used as the RAIN_serial value.
- Be padded to an 8-bit (byte) string with leading zero-bits.

MB11 (UserMem) encoding directives:

- Encoding data to MB11 (UserMem) is optional.
- The MB11 data encoding shall start with a DSFID.

MB11 DSFID 0x24 encoding directives:

- The DSFID 0x24 shall be followed by one of the following control characters:
 - **RVS** indicates there is a RUTvalue following.
 - **DFE** indicates UMdata is following.
- UMdata is a hex string. The following apply:
 - UMdata is optional.
 - It is encoded as a bit string following **DFE**.
 - It shall be terminated with the **EOE** when not ending on a byte boundary, the end of memory, or the end of UWC when using Gen2V3. The **EOE** shall not be included in the reported hex string.
 - It shall be present in lowercase-hex

NOTE 2 The word boundary encoding of UserMem ensures UMdata is always on a byte boundary.

The URI Identifier mode control characters (see Annex A)

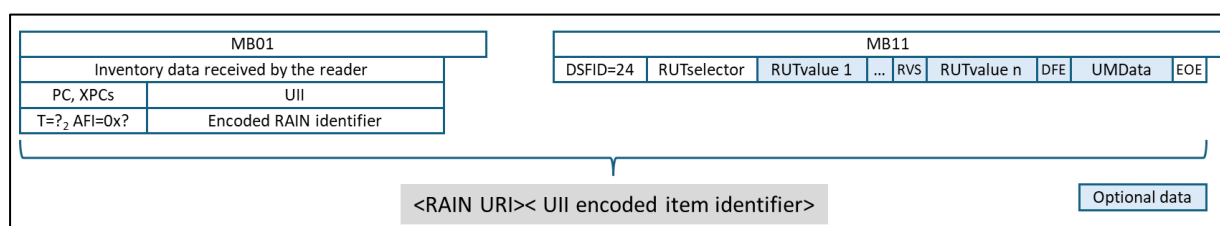
Char	Function
RVS	RUTvalue separator.
SFE	Indicates the Ullserial bit-encoding is following, and there are no valid data in UserMem (MB11).
SFU	Indicates the Ullserial bit-encoding is following, and there are valid data in UserMem (MB11).
STE	Indicates TID shall be used for the RAIN_serial value, and there are no valid data in UserMem (MB11).
STU	Indicates TID shall be used for the RAIN_serial value, and there are valid data in UserMem (MB11).

Char	Function
DFE	Indicates a free-form hex string, UMLdata, is following.
EOE	Indicates the end of the DSFID 0x24 UserMem encoded data. NOTE EOE is the UTF-8 \backslash (0x00) character, as such it supports zero-bit padding.

EXAMPLE See Annex D.

6 URI Add-on mode

The Add-on mode translates the Identifier stored in MB01 to an URI according to the rules of that identifier. In the absence of such rules, the directives of this clause shall be used.



The DSFID 0x24 and AFI not equal to 0xBD indicates that the RAIN URI Identifier is used in its Add-on mode. The RAIN URI Identifier is encoded in MB11 (UserMem).

- Readers may ignore MB11 and use the RAIN item identifier as is, as before.
- 4. Readers should check UserMem for an encoded DSFID and act accordingly, e.g. execute the RAIN URI Identifier Add-on when required by the read scenario. This check can be achieved as follows:
 - Gen2V2: When the UMI bit is set in the PC word then the tag has a MB11 (UserMem). The reader should read only two words from UserMem since many tags is limited to two words in MB11. Check the DSFID and act accordingly.
 - Gen2V3: When the RUM bit is set read UserMem with the ReadVar(0,0) command. This command reads all the UserMem data as indicated by the tag configured UWC (UserMem Word Count). This ensures that the reader reads the intended UserMem data.

The following apply to the URI:

- RAIN_serial shall not be reported.
- RAIN_UUI shall be reported, either in the query or mapped in the path as specified by the selected RUT.
- RAIN_AFI shall be reported.
- RAIN_UMdata shall be reported when encoded.

The following apply to the encoding:

- UTF-8 byte encoding shall be used.

- The DSFID is followed by the RUTselector.
- There may be zero or more RUTvalues.
- RUTvalues shall be separated with the RVS character.
- Additional hex data (UMdata) may be encoded as the last value indicated by the DFE character.
- The encoding is terminated with the EOE character, the end of MB11 memory or the last word of the UWC.

The URI Add-on mode control characters (see Annex A)

Char	Function
RVS	RUTvalue separator.
DFE	Indicates a free-form hex string, UMdata, is following. UMdata terminates with the end of MB11 memory, as indicated in Gen2V3 with UWC, the EOE character.
EOE	Indicates the end of the DSFID 0x24 UserMem encoded data. NOTE EOE is the UTF-8 '\u' (0x00) character, as such it supports zero-bit padding.

EXAMPLE 1 The UII is mapped to the path.

RUT: 'https://id.*.io/{RAIN_UII}'

Encoding:

UII is the RAIN Alliance ISO Number TEXT 'RAIN0123456789'.

RUTvalue: 'rain'

URI: 'https://id.rain.io/d2c1c94e30313233343536373839?RAIN_AFI=ae'

EXAMPLE 2 The UII is not mapped to the path.

RUT: 'https://id.*.io/{RAIN_UII}'

Encoding:

UII is the RAIN Alliance ISO Number TEXT 'RAIN0123456789'.

RUTvalue: 'rain'

URI: 'https://id.rain.io?RAIN_UII=d2c1c94e30313233343536373839&RAIN_AFI=ae'

7 Tag access and query keys

7.1 Keys and key values

The query keys and their values:

#	Key	Description	The query key value
1	RAIN_serial	The TID or the UII encoded serial number UIIserial.	Lowercase-hex number in bytes.
2	RAIN_TID	The full TID.	Lowercase-hex number in bytes.
3	RAIN_UII	Only used in the Add-on mode. The full contents of the UII is placed in the URI.	Lowercase-hex string on a 2-byte boundary.
4	RAIN_UM	Only used in the Identifier mode when the DSFID is not 0x24. The full contents of the UM as indicated by the DSFID.	Lowercase-hex string on a 2-byte boundary.
5	RAIN_UMdata	Only used when the DSFID is not 0x24. The contents of the UM as indicated by the encoding.	Lowercase-hex string on a 2-byte boundary.
6	RAIN_AFI	Only used in the Add-on mode. The AFI as received during Inventory.	A lowercase-hex byte.
7	RAIN_PC	The PC, XPC_W1 and XPC_W2 words as received during Inventory.	Lowercase-hex string on a 2-byte boundary.
8	RAIN_sensoralarm	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
9	RAIN_tagnotification	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
10	RAIN_hazmat	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
11	RAIN_nonremoveable	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
12	RAIN_killable	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
13	RAIN_snapshotsensors	Added to the URI when XPC flag for it is set.	The boolean value 'true'.
14	RAIN_authchallenge	Tag authentication crypto challenge as generated by the reader.	Lowercase-hex string on a 2-byte boundary.
15	RAIN_authresponse	Tag authentication crypto response as generated by the tag.	Lowercase-hex string on a 2-byte boundary.
16	RAIN_readtimestamp	ISO 8601 timestamp of the instance when the tag was read.	The basic ISO 8601 time format.

NOTE 1 The RAIN URI query keys are prefixed with 'RAIN_'.

NOTE 2 The query keys can be listed in any order.

7.2 RAIN_serial

Format: RAIN_serial=<lowercase-hex number in bytes>

RAIN_serial shall be reported either in the query or mapped in the path as specified by the selected RUT.

RAIN_serial may have the value of

- the TID word 0 through to the end of the serialization section of MB10, or
- the UII encoded serial number UIIserial,

as indicated by the encoding control characters.

NOTE 1 This part of the TID can be read using the Gen2v3 ReadVar(TID, 0, 0) command.

NOTE 2 This part of the TID can be longer than the current 96 bits commonly used by commodity tags.

7.3 RAIN_TID

Format: RAIN_TID=<The full TID in lowercase-hex as a byte string>

RAIN_TID should be reported.

NOTE 1 Some standard RAIN identifier encodings, like ISO/IEC 17360, can use the TID.

NOTE 2 The TID may contain information which is important to the application.

7.4 RAIN_UII

Format: RAIN_UII=<the UII as received during Inventory in lowercase-hex>

RAIN_UII shall not be reported in the Identifier mode.

RAIN_UII shall be reported in the Add-on mode, either in the query or mapped in the path as specified by the selected RUT.

7.5 RAIN_UM

Format: RAIN_UM=<DSFID specified UserMem data in lowercase-hex>

RAIN_UM shall not be reported in Add-on mode.

RAIN_UM should be reported in Identifier mode when the DSFID is not 0x24:

Reading the UserMem involves:

- Gen2v2 readers: When the UMI-bit (User Memory Indicator) is set in the PC word received during Inventory, and the reader is configured to read User Memory, then the reader shall read a first part of the User Memory, which is at least one word long.
- Gen2v3 and tags: When the RUM-bit (Read User Memory) is set, then the reader should read the User Memory with the ReadVar(0,0) command. The tag then responds with the User Memory words as set by the UWC (User Memory Word Count).



7.6 RAIN_UMdata

Format: RAIN_UMdata=<data in lowercase-hex>

RAIN_UMdata shall be reported when encoded in MB11 (UserMem), see clauses 5 and 0.

7.7 RAIN_AFI

Format: RAIN_AFI=<AFI as reported during inventory as party of the PC word>

RAIN_AFI shall be reported in the Add-mode when RAIN_PC is not reported.

RAIN_AFI shall not be reported in the Identifier mode.

7.8 RAIN_PC

Format: RAIN_PC=<PC, XPC_W1 and XPC_W2 words as one lowercase-hex string on a 2-byte boundary.>

A reader shall report RAIN_PC when it does not interpret and appropriately report the tag status flags and snapshot sensor.

NOTE The PC and XPC words contains can contain important information about the tag and the item it is attached to.

7.9 RAIN_sensoralarm

RAIN_tagnotification

RAIN_hazmat

RAIN_nonremoveable

RAIN_killable

Format: RAIN_sensoralarm=true

Format: RAIN_tagnotification=true

Format: RAIN_hazmat=true

Format: RAIN_nonremovable=true

Format: RAIN_killable=true

When the XPC word 1 (XPC_W1) is reported by the tag and any of these flags are set, then the appropriate keys shall be reported, unless reported with RAIN_PC.

For the flag specification see ISO/IEC 18000-63 and the RAIN Alliance Reader Communications Interface (RCI).

NOTE 1 When T=0, then these flags can be mapped to the PC word, see GS1 Gen2.

NOTE 2 If present, then the XPC is transmitted by the tag with the UII during Inventory.

7.10 RAIN_snapshotsensors

Format: RAIN_snapshotsensors=<XPC_W2>



When the SnapShot sensor bit is set in XPC_W1 and XPC_W2 is reported then this key shall be appended to the query, see ISO/IEC 18000-63 and the RAIN Alliance Reader Communications Interface (RCI).

The reader may append the specific sensor values as keys in the query as specified by RAIN Alliance Reader Communications Interface (RCI).

NOTE If present, then the XPC is transmitted by the tag with the UII during Inventory.

7.11 RAIN_authchallenge

RAIN_authresponse

Format: RAIN_authchallenge=<tag authentication crypto challenge>

Format: RAIN_authresponse=<tag authentication crypto response >

KeyID=0 shall be used.

7.12 RAIN_readtimestamp

Format: RAIN_readtimestamp=<The time instance of reading the tag in the basic ISO 8601 format>

The reader should add a read timestamp to the URI using this key and format.

Annex A (normative)

The ISO/IEC 15962 6-bit character set as used by the RAIN URI Identifier

The RAIN URI Identifier uses the adapted ISO/IEC 15962 6-bit character set, as specified in the table, for encoding data in MB01 (UII/EPC).

NOTE 1 The ISO/IEC 15962 6-bit character set is limited to uppercase characters, digits and a special-character subset. MB01, for this identifier, uses lowercase letters.

NOTE 2 The 6-bit character set is widely used by automated identification item identifier encodings, like ISO/IEC 15434 (barcodes) and ISO/IEC 17360 (RAIN) item identifiers.

6-bit characters for MB01 encoding

6-bit #	Character	6-bit #	Character	6-bit #	Character	6-bit #	Character
100000	[space] → RFU	110000	0	000000	@ → EOE	010000	P → p
100001	!	110001	1	000001	A → a	010001	Q → q
100010	" → RFU	110010	2	000010	B → b	010010	R → r
100011	# → RFU	110011	3	000011	C → c	010011	S → s
100100	\$	110100	4	000100	D → d	010100	T → t
100101	% → RFU	110101	5	000101	E → e	010101	U → u
100110	&	110110	6	000110	F → f	010110	V → v
100111	'	110111	7	000111	G → g	010111	W → w
101000	(111000	8	001000	H → h	011000	X → x
101001)	111001	9	001001	I → i	011001	Y → y
101010	*	111010	:	001010	J → j	011010	Z → z
101011	+	111011	;	001011	K → k	011011	[→ SFE
101100	,	111100	< → STE	001100	L → l	011100	\ → RFU
101101	-	111101	=	001101	M → m	011101] → SFU
101110	.	111110	> → STU	001110	N → n	011110	^ → RVS
101111	/	111111	? → RFU	001111	O → o	011111	_

Shaded characters are not allowed as base URI, but some shaded characters are allowed as the URI Extension (see B.3).

MB11 (UserMem) control characters

Char	UTF-8	Hex	UTF-8 character name	Function
RVS	␣	0x1F	Unit separator	RUTvalue separator.
DFE	␣	0x1D	Group separator	Indicates a free-form hex string, UMdata, is following.
EOE	␣	0x00	null	Indicates the end of the DSFID 0x24 UserMem encoded data.

Annex B (normative)

The RAIN URI Template (RUT)

B.1 General

A RUT is identified by the RUTselector.

The RUTselector is the first character of the encoding. It is therefore limited to 64 unique values in MB01 (6-bit encoding) and 256 values in MB11 (8-bit encoding).

NOTE This constraint arises from the 96-bit size limitation of commodity RAIN tags, which necessitates the use of a 6-bit character set in MB01.

As the administrator and owner of the list, the RAIN Alliance must exercise prudence in allocating RUTs to ensure long-term sustainability and interoperability.

B.2 The RUT format

The RUT is a URI string containing placeholders to be replaced by data encoded to the tag. The placeholders are:

- The asterisk-character (*) shall be used as the placeholders for the RUTvalues.
 - There may be more than one RUTvalue.
 - The RUTvalues shall be populated in sequence of occurrence and encoding.
- {RAIN_serial} shall be used as the placeholder to map the RAIN_serial value to the URI path. RAIN_serial can be the RAIN_TID_serial or the RAIN_UII_serial (optionally encoded in MB01).
- {RAIN_UII} shall be used as the placeholder to map the UII, in the URI Add-on mode, to the URI path.

EXAMPLE 1 The UII is encoded with '000110_{2rainRVSpa}rtSTE_{6-bit text}' and the TID is E28FF79A2000378BEA406506_{hex}. 'https://id.*.io/*' requires two RUTvalues. The RUTvalues are 'rain' and 'part'. The resulting URI is:

```
'https://id.rain.io/part?RAIN_TID_serial=e28ff79a2000378bea406506'
```

EXAMPLE 2 The UII is encoded with '000111_{2rainRVSpa}rtSTE_{6-bit text}' and the TID is E28FF79A2000378BEA406506_{hex}. RUT 7, 'https://id.*.io/{RAIN_serial}', maps the serial number in the path.

```
'https://id.rain.io/part/e28ff79a2000378bea406506'
```



EXAMPLE 3 The Ull is encoded with XYZ. The RUT 'https://id.*.io/*/{{RAIN_Ull}}' can only be used in the URI Add-on mode where it maps the Ull, as encoded in MB01, in the path. In this example the Ull is a RAIN Alliance ISO Number which is allocated the AFI 0xAE.

'https://id.rain.io/part/d2c1c94e30313233343536373839?RAIN_AFI=ae'

NOTE 1 The AFI must be shown to allow for the resolver to know which Ull is presented (see Annex F).

NOTE 2 The Ull encoding directives of ISO/IEC 18000-63 ensures that the encoded Ull are always on a 16-bit word boundary, i.e. 2-byte boundary. The specific Ull encoding is specified by its specification, which can use any form of padding, typically trailing zero-bit padding since Ulls are specified to be a bit-string.

B.3 The RUTlist format

The RUTlist shall be publicly available in both human-readable and machine-readable formats.

The machine-readable format shall use JSON with the following minimum required fields/keys (non-compliant JSON formatting is shown below for readability):

```
{ "RAIN_URI_Templates":
  { "issue_date": "<ISO date>",
    "Copyright": "The RAIN Alliance - https://therainalliance.org",
    "License": "Free to use unmodified for your own purposes.
               Provided as-is, without warranty or liability.",
    "RUTs":
      [ { "selector": <number>, "RUT": "<RUT>", "instructions": <optional instructions> },
        // more RUT objects
      ]
  }
}
```

B.4 Governance

RUTs shall be free and open for use.

The RUTlist shall be managed within the rules and directives of the RAIN Alliance.

B.5 An example RUTlist

NOTE 1 The RUTlist is managed by the RIN Alliance. As such this list is illustrative of what it can be.

NOTE 2 The RUTs 64 to 255 can only be used in the Add-on mode.

NOTE 3 RAIN_Ull mapping can only be used in the Add-on mode.

EXAMPLE

RUTselector	RUT	Instructions
0	https://*	
1	https://*/{{RAIN_serial}}	The RAIN_serial can be the Ullserial or TID.

RUTselector	RUT	Instructions
2	https://id.*.io	
3	https://id.*.io/{RAIN_serial}	The RAIN_serial can be the UIIserial or TID.
4	https://id.*	
5	https://id.*/{RAIN_serial}	The RAIN_serial can be the UIIserial or TID.
6	https://id.*.io/*	
7	https://id.*.io/*/{RAIN_serial}	The RAIN_serial can be the UIIserial or TID.
8 to 63	Reserved	
64	https://id.rain.io	Limited to URI Add-On mode where the RUT_selector is 8 bits.
65	https://id.rain.io/uii/{RAIN_UII}	Limited to URI Add-On mode where the RUT_selector is 8 bits.
66	https://id.well-known-resolver.io	An example for well-known Add-on mode resolvers.
67 to 255	Reserved	Limited to URI Add-On mode where the RUT_selector is 8 bits.

Annex C (informative)

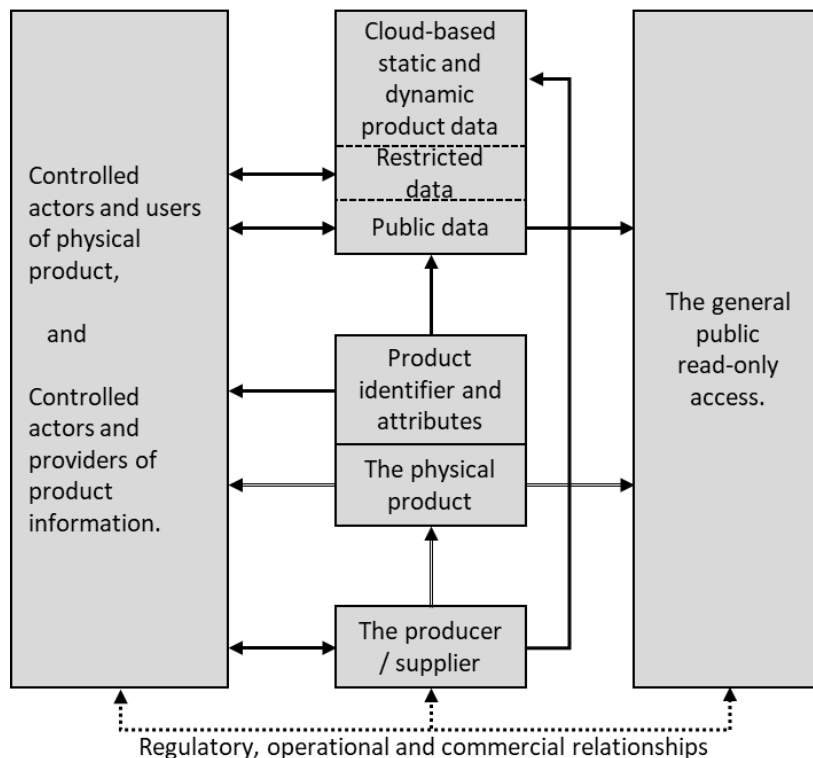
Use of the RAIN URI Identifier as a Digital Product Passport (DPP) Identifier

This annex provides context for the use of the RAIN URI Identifier in its role as a Digital Product Passport (DPP) identifier.

A Digital Product Passport (DPP) is a structured set of [regulated] information associated with a product. This information may include certifications, inspection records, status updates, and usage data relevant to consumers and operators such as point-of-sale systems, logistics providers, reuse and recycling facilities. Although the DPP may be distributed across multiple federated services, it is presented as a cohesive set of data. It may contain both public and restricted information, comprising static elements as well as dynamic components updated by product handlers and producers or suppliers.

The primary purpose of DPP services is to provide both the public and authorized product handlers with the information necessary to manage the product throughout its entire lifecycle. This includes use, selection, shipment, consumption, transformation, support, maintenance, certification, disassembly, recycling, and reuse. DPPs also serve to ensure accountability by producers and authorized handlers for both the product itself and its associated information.

Throughout its lifecycle, a physical product may be owned or handled at various stages by the producer, supplier, authorized handlers, or members of the public.



NOTE The solid lines depicts data flows.

RAIN supports features such as item usage indication and item identifier authentication. The RAIN URI Identifier enhances these capabilities by encoding these additional information in the query portion of the URI.

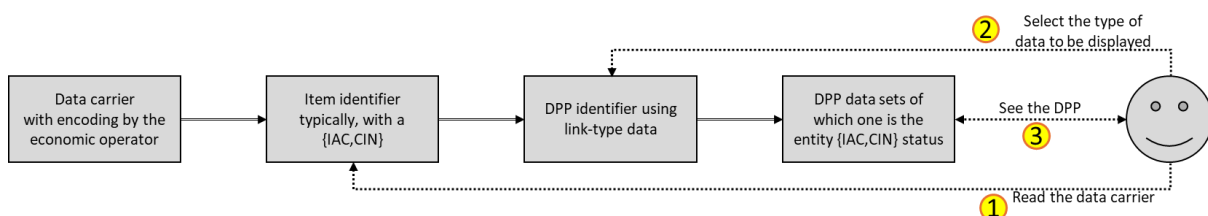
When used in its identifier mode, the RAIN URI Identifier can function as both a product identifier and a DPP identifier. While traditional manufacturing, logistics, and retail systems typically rely on established item or product identifiers, retrofitting such systems to support new identifiers is often impractical.

To address this, the RAIN URI Identifier supports an add-on mode, where it is encoded in the RFID tag's User Memory. This allows it to augment the classic identifier stored in Memory Bank 01 (MB01), such as the UII or EPC. In this mode, legacy systems can continue operating unchanged, while newer DPP-aware systems—such as consumer smartphones—can read both the classic identifier and the RAIN URI Identifier, thereby deriving a DPP identifier without disrupting existing infrastructure.

DPP	Product information
DPP (URI) identifier	Unique product identification, including attributes, sensors and digital security marks
Item (Product) identifier	
Data carrier construct / encoding identifier	
Data carrier (barcodes, NFC, RAIN)	
Product	Physical object with physical security marks

Resolvers are typically responsible for translating item identifiers into DPP identifiers. In consumer or operator applications, the resolver may present options to the user for selecting specific types of DPP information to retrieve, such as usage instructions or recycling guidelines. This selection is commonly encoded using a query parameter, such as linktype.

NOTE Both GS1 and the EU DPP initiative are in the process of standardizing such parameters, including linktype and other DPP identifier keys.



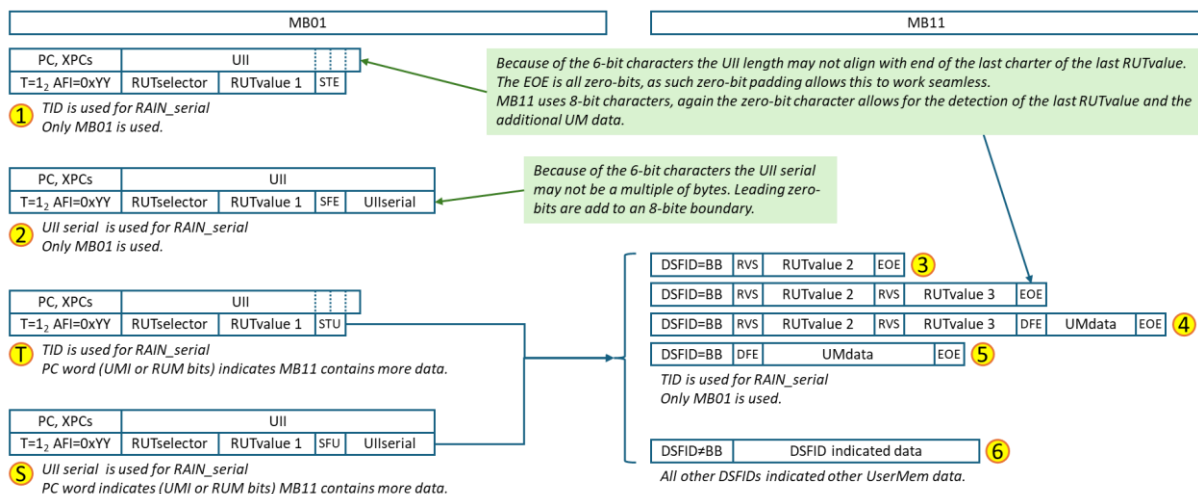
When the RAIN URI Identifier is used in add-on mode, the consumer's device (e.g. a smartphone) can interact with the operator to append query parameters like linktype, thereby enabling direct access to specific product information.

Annex D (informative)

Identifier mode encoding illustrations and examples

D.1 URI Identifier mode

The diagram illustrates the various Identifier mode encoding combinations.



EXAMPLE 1 Only MB01 is used for the encoding. One RUTvalue. TID is used for RAIN_serial.

RUT: 'https://id.*.io'

RUTvalue: 'some_entity'

URI: 'https://id.some_entity.io?RAIN_serial=e28ff79a2000378bea406506'

'some_entity' can be 14 characters long in a 96-bit tag, with this RUT.

EXAMPLE 2 Only MB01 is used for the encoding. One RUTvalue. Ullserial is used for RAIN_serial.

RUT: 'https://id.*.io'

RUTvalue: 'entity'

URI: 'https://id.entity.io?RAIN_serial=0123456789ab'

The RUTselector, RUTvalue and SFE are 8 6-bit characters which use 48 bits. The Ullserial is therefore, in a 96-bit tag 48 bits long, a rather big number range: zero to 281,474,976,710,655.

EXAMPLE 3 with (S) MB01 and MB11 are used for the encoding. Two RUTvalues. Ullserial is used for RAIN_serial. A linktype is added in UserMem to be used when consumer device reads the tag.

RUT: 'https://id.*.io/{RAIN_serial}?linktype=*'



RUTvalue: 'some_entity' and 'DPP'

URI: 'https://id.some_entity.io/0123456789ab?linktype=DPP'

With this encoding when doing only an inventory is done, i.e. only the UII in MB01 is read, then the URI will still resolve properly. This may be done in operational environments. When a consumer device reads the tag, then UserMem should be read, in which case the URI can take the consumer directly to the DPP.

EXAMPLE 4 with (S) MB01 and MB11 are used for the encoding. Three RUTvalues. Ullserial is used for RAIN_serial. UMdata is also added. This is an extension of example 3.

RUT: 'https://id.*.io/iaccin/*/{RAIN_serial}?linktype=*'

RUTvalue: 'some_entity', 'qcidet' and 'DPP'

UMdata: 'ABCD1234_{hex}'

URI: 'https://id.some_entity.io/iaccin/qcidet/0123456789ab?linktype=DPP&RAIN_UMdata=abcd1234'

EXAMPLE 5 with (S) MB01 and MB11 are used for the encoding. One RUTvalue. Ullserial is used for RAIN_serial. MB11 contains only UMdata.

RUT: 'https://id.*.io'

RUTvalue: 'entity'

UMdata: 'ABCD1234_{hex}'

URI: 'https://id.entity.io?RAIN_serial=0123456789ab&RAIN_UMdata=abcd1234'

EXAMPLE 6 with (S) MB01 and MB11 are used for the encoding. One RUTvalue. Ullserial is used for RAIN_serial. MB11 contains data not identified with DSFID 0x24.

RUT: 'https://id.*.io'

RUTvalue: 'entity'

UserMem (MB11): 'ABCDEF0123456789_{hex}'

URI: 'https://id.entity.io?RAIN_serial=0123456789ab&RAIN_UM=abcdef0123456789'

D.2 URI Add-on mode

Two examples are provided, one to fit a 128-bit commodity tag and another with larger memory.

In both case the UII is a RAIN Alliance ISO Number. For it $T=1_2$ and $AF=AE_{hex}$. The UII encoded value is 'c1c2c3443031323334353637_{hex}' which is 'ABCD01234567' in text where "ABCD" is the RAIN Alliance CIN "XRA ABCD", which is assigned to the entity "ABCD Inc." who holds the domain names "abcd-inc.com" and "abcd-inc.io".

The DSFID in MB11 is set to 0x24.

EXAMPLE 1 A commodity tag with a 96-bit UII (in MB01) and a 32-bit UserMem (MB11) is used.



In this example there is only 32 bits available to encode the RAIN URI Identifier it.

As such the RUT [64, 'https://id.rain.io'] is used. This is a resolver which knows what to do with RAIN Alliance ISO number encodings. Therefore there is no RUTvalue, just a RUT selector.

The encoding is:

MB01 – 96 bits	
Inventory data received by the reader	
PC, XPCs	UII
T=1 ₂ AFI=0AE	d2c1c94e3031323334353637

MB11 – 32 bits			
DSFID	RUTselector	EOE	Zero-bit padding
BB	0100 0001	0000 0000	0000 0000

In the PC word UMI=1₂ (Gen2V2), or RUM=1₂ (Gen2V3) and UWC=2, which is the default in many cases.

URI: 'https://id.rain.io?RAIN_UII=c1c2c3443031323334353637&RAIN_AFI=ae'

The RUT [65, 'https://id.rain.io/uii/{RAIN_UII}'] maps the UII to the path.

URI: 'https://id.rain.io/c1c2c3443031323334353637?RAIN_AFI=ae'

NOTE The AFI must be reported to ensure the resolver knows which RAIN item identifier is encoded.

EXAMPLE 2 A tag with a large UserMem (MB11) is used.

UserMem is large enough to be encoded with the identifier issuer resolver URI, which is applicable since a RAIN Alliance ISO number is application specific.

The RUT [2, 'https://id.*.io'] can be used.

The RUTvalue is 'abcd-inc'.

The encoding is:

MB01 – 96 bits	
Inventory data received by the reader	
PC, XPCs	UII
T=1 ₂ AFI=0AE	d2c1c94e3031323334353637

MB11 – many bits				
DSFID	RUTselector	RUTvalue 1	EOE	Zero-bit padding
BB	0000 0010	abcd-inc _{utf-8}	0000 0000	0000 0000

In the PC word UMI=1 (Gen2V2), or RUM=1 (Gen2V3) and UWC=6.

URI: 'https://id.abcd-inc.io?RAIN_UII=c1c2c3443031323334353637&RAIN_AFI=ae'

The RUT [xx, 'https://id.*.io/uii/{RAIN_UII}'] maps the UII to the path.

URI: 'https://id.abcd-inc.io/uii/c1c2c3443031323334353637?RAIN_AFI=ae'

Annex E (informative)

Recommended tag access and reporting

E.1 RAIN Item Identifier Read Scenarios

RAIN Item Identifiers are typically read in two overlapping scenarios: operational reads and Digital Product Passport (DPP) interactions.

5. Operational read scenarios involve infrastructure-based reading of RAIN RFID tags in industrial and commercial environments such as manufacturing, logistics, retail, and access control. These reads are typically automated, managed by local systems, and occur in machine time. They often involve bulk or unattended reads and are usually time-critical.
6. DPP interaction scenarios are initiated to retrieve, update, or verify Digital Product Passport (DPP) data or equivalent structured information associated with an item. These reads occur in human time and may take place in consumer, enterprise, or regulatory contexts. They often require online access to external information sources.

NOTE The term Digital Product Passport (DPP) is used broadly here to include structured digital representations not only of products, but also of any item identified by a RAIN RFID tag, including components, assets, and regulated materials.

E.2 Operational read scenarios

In operational scenarios, read speed is critical, and the controlling application typically knows which tags to read and what data to extract.

7. The RAIN URI Identifier can be reported uninterpreted, i.e. in hexadecimal form. In such cases:
 - The PC word and any XPC words are needed to enable correct interpretation of the UII.
 - The TID is typically needed by the application. Without it, specifically in Identifier mode, multiple tags can appear identical due to shared encodings. To mitigate this, item identifier issuers can use the UIIserial field.
 - In Identifier mode, the RAIN_serial must be included as part of the URI. It does not apply in Add-on mode, as item serialization is already part of the primary item identifier stored in the UII.
 - In Add-on mode, the AFI (Application Family Identifier) is needed by the application to determine the type of encoded item identifier in UII.



- The PC word, including any XPC words received during inventory, is important to for the application. It is received during Inventory and therefore is available to be reported without additional tag access.
- Additional tag access (e.g., user memory or tag authentication) can be performed and reported, depending on application requirements.

E.3 DPP interaction scenarios

A primary goal of the RAIN URI Identifier is to enable consumers and manual operators to interact with the online information associated with the item, similar to how URI-based barcodes and NFC tags are used. It provides access to the advanced features of RAIN RFID without requiring a dedicated app — relying instead on native operating system capabilities.

To support these advanced features on consumer devices, the following applies:

- The RAIN_serial shall be reported in the Identifier mode as part of the URI and it shall not be reported in the Add-on mode.
- The AFI shall be reported in the Add-on mode as part of the URI and it shall not be reported in the Identifier mode.
- The TID should be read and reported as part of the URI.
- The PC word, including the XPC words when present, transmitted during inventory, shall be reported as part of the URI.
- The first part of User Memory (MB11), containing the DSFID, should be reported as part of the URI. If the DSFID is 0x24, then the reader should append the extended URI to the URI.
- Tag-use flags, if set with T=0 and mapped to the PC word or provided with XPC_W1, should be interpreted as specified in the RAIN Alliance Reader Communication Interface (RCI) and reported as part of the URI.
- Snapshot sensor data, if supplied with the XPC W2, may be interpreted as specified in the RAIN Alliance Reader Communication Interface (RCI) and reported as part of the URI.
- RAIN on-chip crypto tag authentication, when using the identifier mode, should be attempted using the Proxy method as specified in RAIN Alliance Reader Communication Interface (RCI) and reported as part of the URI, when successful.

NOTE 1 Encryption of data is out of scope because of the key management complexities.

NOTE 2 Reading of a RAIN URI Identifier typically happens in "human time". The reader-tag transaction time (tag interrogation time), and message-packet-size are in this case not a limiting factor. Readers in operational lines can be configured to read only what it needs, i.e. according to the application-read-scenario.