Base64 Encoding Algorithm

This section describes the Base64 encoding algorithm with some simple encoding examples.

Base64 algorithm is designed to encode any binary data, an stream of bytes, into a stream of 64-printable characters.

Base64 encoding algorithm was first presented in "RFC 1421 - Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures" in 1993 by John Linn. It was later modified slightly in "RFC 1521 - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies" in September 1993 by N. Borenstein, et al..

The character set map by Base64 encoding:

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

The Base64 encoding process is to:

- Divid the input bytes stream into blocks of 3 bytes.
- Divid 24 bits of each 3-byte block into 4 groups of 6 bits.
- Map each group of 6 bits to 1 printable character, based on the 6-bit value using the Base64 character set map.
- If the last 3-byte block has only 1 byte of input data, pad 2 bytes of zero (\x0000). After encoding it as a normal block, override the last 2 characters with 2 equal signs (==), so the decoding process knows 2 bytes of zero were padded.
- If the last 3-byte block has only 2 bytes of input data, pad 1 byte of zero (\x00). After

encoding it as a normal block, override the last 1 character with 1 equal signs (=), so the decoding process knows 1 byte of zero was padded.

- Carriage return (\r) and new line (\n) are inserted into the output character stream. They will be ignored by the decoding process.

Example 1: Input data, 1 byte, "A". Encoded output, 4 characters, "QQ=="

```
Input Data          A
Input Bits    01000001
Padding       01000001 00000000 00000000
                     \       \       \
Bit Groups    010000 010000 000000 000000
Mapping            Q      Q      A      A
Overriding         Q      Q      =      =
```

Example 2: Input data, 2 bytes, "AB". Encoded output, 4 characters, "QUI="

```
Input Data          A        B
Input Bits    01000001 01000010
Padding       01000001 01000010 00000000
                     \       \       \
Bit Groups    010000 010100 001000 000000
Mapping            Q      U      I      A
Overriding         Q      U      I      =
```

Example 3: Input data, 3 bytes, "ABC". Encoded output, 4 characters, "QUJD"

```
Input Data          A        B        C
Input Bits    01000001 01000010 01000011
                     \       \       \
Bit Groups    010000 010100 001001 000011
Mapping            Q      U      J      D
```

Submit Your Comment:

Please write meaningful comments. Thanks ☺

Your Name:

Let (a,b)=(9,2), what is a-b?

(All fields are required.) Submit

Table of Contents