# BAS32$ - Convert Binary String to Base32

The BAS32$ routine can be used to convert a block of binary data to an ASCII string by translating it to a Radix-32 representation.

## 1. Invocation

To convert a string to Base32 either code:

```
CALL BAS32$ USING input in_len output out_len
```

or:

```
CALL BAS32$ USING b32
```

The subroutine call with 4 parameters is compatible with the BAS64$ calling convention) where *input* is the binary string to be converted, *in_len* is a PIC 9(4) COMP, or literal, that must contain the length of the input field; *output* is a buffer which will contain the resulting ASCII string; and *out_len* is a PIC 9(4) COMP field which will be updated to hold the length of the output string.

By default, the output string will be padded to an exact multiple of 8 characters (see below). A binary string of *N* bytes will be converted to an ASCII string of 8 * [(*N*+4) / 5(rounded)] characters so the output field must be large enough to hold the resulting string. For example, an input block of 20 bytes will be converted to an ASCII string of 32 characters.

The subroutine call with a single parameter provides full control of the output string where *b32* is a control block of the following structure:

```
01    B32
  02  B32VER      PIC 9(4) COMP          * CONTROL BLOCK VERSION NUMBER
      VALUE       1                      * MUST BE 1
  02  B32ILEN     PIC 9(4) COMP          * LENGTH OF INPUT STRING
  02  B32IPTR     PIC PTR                * POINTER TO INPUT STRING
  02  B32OLEN     PIC 9(4) COMP          * RETURNED LENGTH OF OUTPUT STRING
  02  B32OPTR     PIC PTR                * PTR TO BUFFER FOR BASE32 STRING
  02  B32PADD     PIC X                  * PADDING MODE/CHARACTER
                                         * #20 NO PADDING
                                         * #00 ADD SINGLE #00 TERMINATOR
                                         * #01 DEFAULT PADDING CHAR "="
                                         * #NN PADDING CHARACTER
  02  B32CHAR     PIC PTR                * POINTER TO BASE32 CHARACTER SET
                                         * OR HIGH-VALUES TO USE DEFAULT
```

The length of the output string will depend on the padding mode:

| Input string length | Length of output string | | |
|---|---|---|---|
| | B32PADD = #20 | B32PADD = #00 | B32PADD = #01 or #NN |
| 5n (for n nonzero) | 8n | 8n + 1 | 8n |
| 5n + 1 | 8n + 2 | 8n + 3 | 8n + 8 |
| 5n + 2 | 8n + 4 | 8n + 5 | 8n + 8 |
| 5n + 3 | 8n + 5 | 8n + 6 | 8n + 8 |
| 5n + 4 | 8n + 7 | 8n + 8 | 8n + 8 |

## 2. STOP Codes and Exception Conditions

The following STOP codes may be generated by BAS32$:

| **STOP code** | **Description** |
|---|---|

| | |
|---|---|
| 24124 | The input buffer length was 0. |
| 24125 | The control block version is invalid (The input buffer length was 0. |

No exception conditions are returned by BAS32$.

# 3. Programming Notes

BAS32$ is only available with GSM SP-36 or later.

BAS32$ operates by splitting each group of 5-bytes (i.e. 40-bits) in the input string to 8 groups of 5-bits. Each 5-bit number is used as an index to a 32-entry character table. By default, the character table conforms to the RFC4648 Base32 alphabet:

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZ234567"
```

However, the default Base32 alphabet can be replaced by another set of alphanumeric symbols passed via the B32CHAR pointer. For example:

```
"0123456789ABCDEFGHIJKLMNOPQRSTUV"      Base32hex
"0123456789ABCDEFGHJKMNPQRSTVWXYZ"      Crockford's Base32
"ybndrfg8ejkmcpqxot1uwisza345h769"       z-base-32
```

If a non-zero, non-SPACE padding character has been specified then, if the input string length is not an exact multiple of 5, either one, three, four or six padding characters are appended to the end of the output string. For example:

| **Binary block** | **Base32 string** |
|---|---|
| #01 | AE====== |
| #0102 | AEBA==== |
| #010203 | AEBAG=== |
| #01020304 | AEBAGBA= |
| #0102030405 | AEBAGBAF |
| #010203040506 | AEBAGBAFAY===== |

The default padding character is a "=" but this can be overridden by initialising the B32PADD variable to the desired padding character.

# 4. Examples

[EXAMPLE REQUIRED]

# 5. Copy-Books

None.

# 6. See Also

BAS64$ Convert Binary string to Base-64
BI-HX$ Convert Binary string to Base-16 (hexadecimal)

Note that a DEC32$ sub-routine (convert Base-32 string to binary) is not currently available.