

Dump - Print the contents of binary files in hexadecimal and ATASCII#

General Information

Author: Mark Rose

Language: ACTION!

Compiler/Interpreter: ACTION!

Published: February, 1985

```
; Dump - Print the contents of binary
; files in hexadecimal and ATASCII
```

```
; by Mark Rose - February, 1985
```

```
; A few useful definitions:
```

```
DEFINE ChunkSize = "8",
        Escape    = "$1B",
        NewLine   = "$9B",
        File      = "1"
```

```
; Print out a byte as two hexadecimal
; digits.
```

```
PROC HexByte(BYTE c)
    BYTE ARRAY HexDig(16)=
        ['0 '1 '2 '3 '4 '5 '6 '7
         '8 '9 'A 'B 'C 'D 'E 'F]

    Put(HexDig(c RSH 4))
    Put(HexDig(c&15))
RETURN
```

```
; Print out a two-byte value as 4
; hexadecimal digits by calling
; HexByte.
```

```
PROC HexWord(CARD i)
    HexByte(i RSH 8)
    HexByte(i & 255)
RETURN
```

```
; Read in the next few bytes of the
; file (the desired number is chosen
; by the value of "ChunkSize").
```

```
BYTE FUNC ReadChunk( BYTE ARRAY buf )
    BYTE nBytes,
        c
```

```

nBytes = 0
DO
    c = GetD( File )
    IF EOF( File ) THEN
        EXIT
    FI
    buf( nBytes ) = c
    nBytes ==+ 1
UNTIL nBytes = ChunkSize
OD
RETURN( nBytes )

```

; Put a character to screen. If char
; is an ATASCII return, put period,
; instead, so display isn't messed up.

```

PROC PutChar( BYTE c )
    IF c # NewLine THEN
        Put( Escape )
        Put( c )
    ELSE
        Put( '.' )
    FI
RETURN

```

; Print hex and ATASCII of chars read
; by ReadChunk.

```

PROC DumpChunk( CARD offset, BYTE n, BYTE ARRAY buf )
    BYTE i

    HexWord( offset )
    Print( ":" )
    FOR i = 0 TO n-1
        DO
            HexByte( buf( i ) )
            Put( ' ' )
        OD
    FOR i = i TO ChunkSize-1
        DO
            Print( " " )
        OD
    FOR i = 0 TO n-1
        DO
            PutChar( buf( i ) )
        OD
    PutE()
RETURN

```

; Dump a file to screen.

```

PROC Dump()

```

```
BYTE ARRAY fName( 50 )
CARD offset
BYTE size
BYTE ARRAY buf( ChunkSize )

; First, get file to dump.
Print( "File: " )
InputS( fName )
Close( File )
Open( File, fname ,4 ,0 )

; Until end-of-file, read a few chars
; and dump them to screen.
offset = 0
DO
    size = ReadChunk( buf )
    IF size = 0 THEN
        EXIT
    FI
    DumpChunk( offset, size, buf )
    offset ==+ size
OD

Close(1)
RETURN
```