

Migration Specialties International, Inc.

217 West 2nd Street, Florence, CO 81226-1403

719-784-9196, Fax: 888-854-3417

E-mail: Info@MigrationSpecialties.com

www.MigrationSpecialties.com



CBL Conversion Examples

Attached you will find some examples of VAX DIBOL code converted to VAX C code using the CBL Translator. The code in these examples was converted using the CBL Translator's maximum optimization capabilities. We hope you find these examples useful in reviewing the appearance of the converted C code. Please feel free to contact us if you have any questions.

DIBOL DATA DEFINITIONS

COMMON			
	TTCHN	,D3	; Terminal.
	FLAGS	,D10	; DIBOL flags.
RECORD CURPOS			
		,A1	, '['
	ROW	,D2	; Screen row.
		,A1	, ';'
	COL	,D3	; Screen column.
		,A1	, 'H'
RECORD			
	TROW	,D2	; Top row.
	TCOL	,D3	; Top column.
	TOTAL	,D18	; Total value.
	TTYPER	,D1	; Total numeric type.
	TLNGTH	,D3	; Total length.
	TDECML	,D2	; Total decimal places.
	ENTRY	,D18	; Entered value.
	ETYP	,D1	; Entered numeric type.
	ELNGTH	,D3	; Entered length.
	EDECML	,D2	; Entered decimal places.
	AENTRY	,A14	; Alpha entry.
	AENTL	,D2	; Alpha entry length.
	VALUE	,D18	; Decimal length.
	AVALUE	,A20	; Full alpha value.
	AVALN	,D3	; and length.
	DISPLAY	,A14	; Window display field.
	CHARS	,D3	; Character count.
	FNCTYP	,D2	; Function type.
	FNCODE	,D3	; Function code.
	NTYP	,D1	; Numeric type.
	NLNGTH	,D3	; Numeric length.
	NDECML	,D2	; Numeric decimal places.
	WROW	,D2	; Work row.
	WCOL	,D3	; Work column.
	MATHOP	,D1	; Required operation.
	OPCODE	,D1	; Operation code.
	POINT	,D1	; Decimal point flag.
	NEGTIV	,D2	; Negative value flag & position.
	MISS	,D1	; Miss character flag.
	SGR	,D4	; Graphic rendition.

IFV	,D4		; Input field video.
DFV	,D4		; Display field video.
RECORD	AFLD		; Work field.
	DFLD	,D18	
		,A62	
RECORD			
	NINES	,D18	,999999999999999999
	TENS	,D18	,100000000000000000
LITERAL			
	ESC	,D2	,27
	EQUALS ,D1	,0	
	ADD	,D1	,1
	SUB	,D1	,2
	MUL	,D1	,3
	DIV	,D1	,4
	DBLESC ,D1	,4	
	PF	,D1	,6
	KEYPAD ,D1	,7	
	FUNCTN ,D1	,8	
	PF1	,D3	,256
	PF2	,D3	,257
	PF3	,D3	,258
	PF4	,D3	,259
	KP0	,D3	,260
	KP1	,D3	,261
	KP2	,D3	,262
	KP3	,D3	,263
	KP4	,D3	,264
	KP5	,D3	,265
	KP6	,D3	,266
	KP7	,D3	,267
	KP8	,D3	,268
	KP9	,D3	,269
	ENTER	,D3	,270
	MINUS	,D3	,271
	COMMA ,D3	,272	
	PERIOD ,D3	,273	
	F17	,D3	,297
	F18	,D3	,298
	F19	,D3	,299
	F20	,D3	,300

CONVERTED C DATA DEFINITIONS

```
static struct {
    char fill0001_[1];
    char row_[2];
    char fill0002_[1];
    char col_[3];
    char fill0003_[1];
} curpos_ =  {{{ '[' }
              ,{' ' }
              ,{' ; ' }
              ,{' ' ' }
              ,{' H ' } } };

static SINT4 trow_;
static SINT4 tcol_;
static char total_[18];
static SINT4 ttype_;
static SINT4 tlength_;
static SINT4 tdecml_;
static char entry_[18];
static SINT4 etype_;
static SINT4 elength_;
static SINT4 edecml_;
static char aentry_[14];
static SINT4 aentl_;
static char value_[18];
static char avalue_[20];
static SINT4 avaln_;
static char display_[14];
static SINT4 chars_;
static SINT4 fnctyp_;
static SINT4 fncode_;
static SINT4 ntype_;
static SINT4 nlength_;
static SINT4 ndecml_;
static SINT4 wrow_;
static SINT4 wcol_;
static SINT4 mathop_;
static SINT4 opcode_;
static SINT4 point_;
static SINT4 negtiv_;
static SINT4 miss_;
static SINT4 sgr_;
static SINT4 ifv_;
static SINT4 dfv_;

static struct {
    char dfld_[18];
    char fill0004_[62];
} afld_;

static char nines_[18] = {'9','9','9','9','9','9','9','9','9','9','9','9','9','9','9','9','9','9','9','9'};

static char tens_[18] = {'1','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'};

#define ESC 27
#define ADD 1
#define SUB 2
#define MUL 3
#define DIV 4
```

```

#define DBLESC 4
#define PF 6
#define KEYPAD 7
#define FUNCTN 8
#define PF1 256
#define PF2 257
#define PF4 259
#define KP0 260
#define KP9 269
#define ENTER 270
#define MINUS 271
#define PERIOD 273
#define F17 297
#define F18 298
#define F19 299
#define F20 300

SHAREABLE VAR(flags, 0, DECML, DSC$K_DTYPE_NZ, 10, flags$)
SHAREABLE RECORD(curpos, 0, ALPHA, DSC$K_DTYPE_T, sizeof(curpos_),
    (char *)&curpos_)
SHAREABLE VAR(row, 0, DECML, DSC$K_DTYPE_NZ, 2, curpos_.row_)
SHAREABLE VAR(col, 0, DECML, DSC$K_DTYPE_NZ, 3, curpos_.col_)
SHAREABLE VAR(total, 0, DECML, DSC$K_DTYPE_NZ, 18, total_)
SHAREABLE VAR(entry, 0, DECML, DSC$K_DTYPE_NZ, 18, entry_)
SHAREABLE VAR(aentry, 0, ALPHA, DSC$K_DTYPE_T, 14, aentry_)
SHAREABLE VAR(value, 0, DECML, DSC$K_DTYPE_NZ, 18, value_)
SHAREABLE VAR(avalue, 0, ALPHA, DSC$K_DTYPE_T, 20, avalue_)
SHAREABLE VAR(display, 0, ALPHA, DSC$K_DTYPE_T, 14, display_)
SHAREABLE RECORD(afld, 0, ALPHA, DSC$K_DTYPE_T, sizeof(afld_), (char *)&afld_)
SHAREABLE VAR(dfld, 0, DECML, DSC$K_DTYPE_NZ, 18, afld_.dfld_)
SHAREABLE VAR(nines, 0, DECML, DSC$K_DTYPE_NZ, 18, nines_)
SHAREABLE VAR(tens, 0, DECML, DSC$K_DTYPE_NZ, 18, tens_)

```

DIBOL PROCEDURE CODE (SIMPLE EXAMPLE)

```

trow = 5
tcol = 5
row = trow
col = tcol + 1

```

CONVERTED C CODE (SIMPLE EXAMPLE)

```

trow_ = 5;
tcol_ = 5;
Asni(&row, trow_);
Asni(&col, tcol_ + 1)

```

DIBOL PROCEDURE CODE (EXTENSIVE EXAMPLE)

```

USING MATHOP SELECT
    (ADD, SUB),
    BEGIN
        IF (MATHOP.EQ.SUB) ENTRY = -ENTRY
        IF (TDECML.LT.EDECML) THEN
            BEGIN
                TOTAL = TOTAL * TENS(1, 1 + (EDECML - TDECML))
                NDECML = EDECML
                TLNGTH = TLNGTH + (EDECML - TDECML)
            END
        END

```

```

ELSE
    BEGIN
        ENTRY = ENTRY * TENS(1, 1 + (TDECML - EDECML))
        NDECML = TDECML
        ELNGTH = ELNGTH + (TDECML - EDECML)
        END
TOTAL = TOTAL + ENTRY
IF (TLNGTH.GT.ELNGTH) THEN
    NLNGTH = TLNGTH
ELSE
    NLNGTH = ELNGTH
IF (NLNGTH.LT.18.AND.NINES(1,NLNGTH).LT.TOTAL) INCR NLNGTH
IF (TOTAL.LT.0) THEN
    NTYPE = 1
ELSE
    NTYPE = 0
END
(MUL),
BEGIN
TOTAL = TOTAL * ENTRY
NTYPE = TTYPE * ETYPE
NDECML = TDECML + EDECML
NLNGTH = TLNGTH + ELNGTH - 1
IF (NLNGTH.GT.18) NLNGTH = 18
IF (NLNGTH.LT.18.AND.NINES(1,NLNGTH).LT.TOTAL) INCR NLNGTH
END
(DIV),
BEGIN
IF (EDECML.LT.TDECML)
&    ENTRY = ENTRY * TENS(1, 1 + (TDECML - EDECML))
IF (TDECML.LT.EDECML)
    BEGIN
        TOTAL = TOTAL * TENS(1, 1 + (EDECML - TDECML))
        TLNGTH = TLNGTH + EDECML - TDECML
        TDECML = EDECML
        END
NDECML = TDECML + 1
IF (NDECML.LT.7) NDECML = 7
    DO
        BEGIN
            NDECML = NDECML - 1
            NLNGTH = TLNGTH - TDECML + NDECML
            END
        UNTIL NLNGTH.LT.18
IF (ENTRY) THEN
    TOTAL = TOTAL * TENS(1, 1 + NDECML) / ENTRY
ELSE
    TOTAL = 0
END
ENDUSING

```

CONVERTED C CODE (EXTENSIVE EXAMPLE)

```

/* using mathop_ select */
if ((mathop_ == ADD) || (mathop_ == SUB)) {
    {
        if (mathop_ == SUB) Asn(&entry, Minus(&entry));
        if (tdecml_ < edecml_) {
            {
                Asn(&total, Mul(&total, Dsbs(&tens, 1, 1 + (edecml_ - tdecml_)));
                ndecml_ = edecml_;
            }
        }
    }
}

```

```
        tlength_ = tlength_ + (edecml_ - tdecml_);
    }
} else
{
    Asn(&entry, Mul(&entry, Dsbs(&tens, 1, 1 + (tdecml_ - edecml_)));
    ndecml_ = tdecml_;
    elength_ = elength_ + (tdecml_ - edecml_);
}
Asn(&total, Add(&total, &entry));
if (tlength_ > elength_) {
    nlength_ = tlength_;
} else
    nlength_ = elength_;
if (nlength_ < 18 && Lt(Dsbs(&nines, 1, nlength_), &total)) ++nlength_;
if (Lt(&total, Itod(0))) {
    ntype_ = 1;
} else
    ntype_ = 0;
}
}
else if (mathop_ == MUL) {
    {
        Asn(&total, Mul(&total, &entry));
        ntype_ = ttype_ * etype_;
        ndecml_ = tdecml_ + edecml_;
        nlength_ = tlength_ + elength_ - 1;
        if (nlength_ > 18) nlength_ = 18;
        if (nlength_ < 18 && Lt(Dsbs(&nines, 1, nlength_), &total)) ++nlength_;
    }
}
else if (mathop_ == DIV) {
    {
        if (edecml_ < tdecml_) Asn(&entry, Mul(&entry, Dsbs(&tens, 1,
            1 + (tdecml_ - edecml_))));
        if (tdecml_ < edecml_)
        {
            Asn(&total, Mul(&total, Dsbs(&tens, 1, 1 + (edecml_ - tdecml_))));
            tlength_ = tlength_ + edecml_ - tdecml_;
            tdecml_ = edecml_;
        }
        ndecml_ = tdecml_ + 1;
        if (ndecml_ < 7) ndecml_ = 7;
        do
        {
            ndecml_ -= 1;
            nlength_ = tlength_ - tdecml_ + ndecml_;
        }
        while (!(nlength_ < 18));
        if (Dtoi(&entry)) {
            Asn(&total, Div(Mul(&total, Dsbs(&tens, 1, 1 + ndecml_)), &entry));
        } else
            Asni(&total, 0);
    }
}
}
else ; /* no default */
/* endusing */
```