

# Technische Manual: Intel C DLL voor Rocket COBOL 11.0 en IFX Fortran (F77-stijl)

## Technische Manual

Intel C → DLL → Rocket COBOL 11.0 → IFX Fortran (F77-stijl)

Datum- en tijdroutine met milliseconden in gedeelde C-structuur

---

### 1. Inleiding

Deze handleiding beschrijft hoe een Intel C-routine wordt gebouwd als Windows x64 DLL die datum en tijd met milliseconden retourneert in een platte C-structuur. Deze DLL wordt vervolgens gebruikt door zowel Rocket COBOL 11.0 als IFX Fortran, inclusief native Fortran 77-stijl code, zodat de Fortran 95 DATE\_AND\_TIME-routine niet meer nodig is.

---

### 2. C-ontwerp

#### 2.1 Structuurdefinitie

De structuur gebruikt uitsluitend 2-byte integers (C short) voor maximale compatibiliteit:

```
typedef struct DATETIME_MS_TAG {  
    short year;  
    short month;  
    short day;  
    short hour;  
    short minute;  
    short second;  
    short millisecond;  
} DATETIME_MS;
```

## 2.2 DLL-routine

```
__declspec(dllexport)
void __cdecl get_datetime_ms(DATETIME_MS *dt)
{
    SYSTEMTIME st;
    if (!dt) return;

    GetLocalTime(&st);

    dt->year = (short)st.wYear;
    dt->month = (short)st.wMonth;
    dt->day = (short)st.wDay;
    dt->hour = (short)st.wHour;
    dt->minute = (short)st.wMinute;
    dt->second = (short)st.wSecond;
    dt->millisecond = (short)st.wMilliseconds;
}
```

---

## 3. DLL-buildproces (Intel C / MSVC-compatibel)

### 3.1 Compileren

```
icl /c datetime.c
```

### 3.2 Linken naar DLL

```
link /DLL datetime.obj /OUT:datetime.dll /IMPLIB:datetime.lib
```

### 3.3 Export controleren

```
dumpbin /EXPORTS datetime.dll
```

---

## 4. Integratie met Rocket COBOL 11.0

### 4.1 COBOL-structuur (COMP-5)

```
01 DT-STRUCT.
05 DT-YEAR PIC S9(4) COMP-5.
05 DT-MONTH PIC S9(4) COMP-5.
05 DT-DAY PIC S9(4) COMP-5.
05 DT-HOUR PIC S9(4) COMP-5.
05 DT-MINUTE PIC S9(4) COMP-5.
05 DT-SECOND PIC S9(4) COMP-5.
05 DT-MILLISECOND PIC S9(4) COMP-5.
```

## 4.2 COBOL-aanroep

```
CALL "get_datetime_ms"  
USING BY REFERENCE DT-STRUCT
```

---

## 5. Integratie met IFX Fortran (F77-stijl)

### 5.1 C-interoperabele struct (klein F90-laagje)

```
MODULE datetime_mod  
USE, INTRINSIC :: ISO_C_BINDING  
IMPLICIT NONE  
  
TYPE, BIND(C) :: DATETIME_MS  
  INTEGER(C_INT16_T) :: year  
  INTEGER(C_INT16_T) :: month  
  INTEGER(C_INT16_T) :: day  
  INTEGER(C_INT16_T) :: hour  
  INTEGER(C_INT16_T) :: minute  
  INTEGER(C_INT16_T) :: second  
  INTEGER(C_INT16_T) :: millisecond  
END TYPE DATETIME_MS  
  
INTERFACE  
  SUBROUTINE get_datetime_ms(dt) BIND(C, NAME="get_datetime_ms")  
    USE, INTRINSIC :: ISO_C_BINDING  
    TYPE(DATETIME_MS) :: dt  
  END SUBROUTINE get_datetime_ms  
END INTERFACE  
END MODULE datetime_mod
```

## 5.2 F77-stijl hoofdprogramma

```
PROGRAM TESTDATETIME
USE datetime_mod
IMPLICIT NONE

TYPE(DATETIME_MS) :: DT

CALL get_datetime_ms(DT)

WRITE(*,'(A,I4.4,"-",I2.2,"-",I2.2)') 'Date: ',
& DT%year, DT%month, DT%day

WRITE(*,'(A,I2.2,":",I2.2,":",I2.2,".",I3.3)') 'Time: ',
& DT%hour, DT%minute, DT%second, DT%millisecond

END
```

## 5.3 Build

```
ifx /c datetime_mod.f90
ifx /c main_f77.f
ifx main_f77.obj datetime_mod.obj datetime.lib /exe:testdatetime.exe
```

---

## 6. Waarom dit Fortran 95-routines vervangt

- Windows API levert milliseconden, wat Fortran 95 standaard niet doet.
- De C-DLL levert een stabiele binaire struct die door F77-code gelezen kan worden.
- Geen afhankelijkheid meer van DATE\_AND\_TIME.

---

## 7. Uitbreidbaarheid

- Extra velden: epochSeconds, timezoneOffsetMinutes, dayOfWeek.
- Extra DLL-export: get\_datetime\_utc\_ms.
- Geformatteerde datum/tijd-string als tweede routine.

---

## 8. Conclusie

Deze manual levert een volledig reproduceerbare, taalafhankelijke, milliseconde-nauwkeurige datum-tijdoplossing die:

- in Intel C wordt gebouwd,

- als DLL wordt gedeeld,
- door Rocket COBOL 11.0 wordt gebruikt,
- door IFX Fortran (F77-stijl) wordt gebruikt,
- zonder afhankelijkheid van Fortran 95 datetime-routines.