

# IFX Fortran <-> Rocket COBOL 11.0 Interoperability

Technisch document - A4-PDF-versie

## 1. Overzicht

Dit document beschrijft hoe IFX Fortran en Rocket COBOL 11.0 op een betrouwbare manier kunnen samenwerken via gedeelde libraries, consistente calling conventions en correcte datatypemapping. De inhoud is geoptimaliseerd voor een A4-PDF-layout.

## 2. Interoperabiliteitsarchitectuur

### 2.1 Aanbevolen integratiemodel

De meest stabiele en performante methode is: COBOL roept Fortran-subroutines aan via gedeelde libraries (DLL/SO).

### 2.2 Alternatieven

- Fortran -> COBOL calls
- Bestandsuitwisseling
- IPC (pipes/sockets)

## 3. Calling Conventions

### 3.1 Naamgeving

IFX Fortran exporteert symbolen als lowercase + trailing underscore. Voorbeeld: SUBX -> subx\_. COBOL-aanroep: CALL "subx\_".

### 3.2 Argument-passing

Beide talen gebruiken pass-by-reference.

### 3.3 Strings

Fortran geeft strings door als pointer + lengte. COBOL gebruikt fixed-length of wrapper.

## 4. Compiler- en Linkinstellingen

IFX Fortran:

```
ifx -shared -fPIC mymath.f90 -o libmymath.so
```

Rocket COBOL:

```
cob -x mainprog.cbl -L. -lmymath
```

## 5. Voorbeeld: COBOL roept Fortran

Fortran:

```
subroutine addxy(x, y, result)
    real(8) :: x, y, result
```

```

        result = x + y
end subroutine addxy

```

```

COBOL:
IDENTIFICATION DIVISION.
PROGRAM-ID. CALL-FORTRAN.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 X-VALUE          COMP-2 VALUE 10.0.
01 Y-VALUE          COMP-2 VALUE 20.0.
01 RESULT-VALUE     COMP-2.
PROCEDURE DIVISION.
    CALL "addxy_" USING X-VALUE Y-VALUE RESULT-VALUE
    END-CALL.
    DISPLAY "Result = " RESULT-VALUE.
    STOP RUN.

```

## 6. Voorbeeld: Fortran roept COBOL

```

COBOL subroutine:
IDENTIFICATION DIVISION.
PROGRAM-ID. MULTIPLYX.
DATA DIVISION.
LINKAGE SECTION.
01 A COMP-2.
01 B COMP-2.
01 R COMP-2.
PROCEDURE DIVISION USING A B R.
    COMPUTE R = A * B.
    EXIT PROGRAM.

```

Fortran:

### interface

```

subroutine multiplyx(a, b, r) bind(C, name="multiplyx")
    real(8) :: a, b, r
end subroutine multiplyx

```

### end interface

### program test

```

real(8) :: a, b, r
a = 3.0
b = 4.0
call multiplyx(a, b, r)
print *, "Result:", r

```

### end program test

## 7. Best-practices

- Gebruik REAL(8) <-> COMP-2 voor numerieke stabiliteit
- Gebruik wrapper-modules voor arrays en structuren
- Inspecteer symbolen met nm
- Gebruik LD\_DEBUG=libs bij library-loading problemen

- Houd naamgeving consistent

## **8. Conclusie**

IFX Fortran en Rocket COBOL 11.0 kunnen uitstekend samenwerken via gedeelde libraries, mits calling conventions en datatypen correct worden toegepast.