

Intel C RNG DLL voor gebruik in Rocket COBOL 11.0

Intel C RNG DLL voor gebruik in Rocket COBOL 11.0

1. Overzicht

Dit document beschrijft hoe je een zelfstandige random number generator (RNG) implementeert als Windows DLL met **Intel C**, en hoe je deze DLL vervolgens aanroept vanuit **Rocket COBOL 11.0**. De interface is ontworpen voor maximale compatibiliteit: voorspelbare exportnaam, __stdcall calling convention en directe mapping naar COBOL datatypen.

2. Intel C RNG DLL (Windows)

De DLL bevat één functie:

- Naam: RNG_NEXT
- Calling convention: __stdcall
- Parameter: pointer naar een double
- COBOL-mapping: COMP-2

2.1 C-broncode (rng.c)

```
#include <stdlib.h>
#include <time.h>

static int rng_initialized = 0;

static void rng_init(void) {
    if (!rng_initialized) {
        srand((unsigned int)time(NULL));
        rng_initialized = 1;
    }
}

__declspec(dllexport) void __stdcall RNG_NEXT(double *value) {
    double r;
    rng_init();
    r = (double)rand() / (double)RAND_MAX;
    *value = r;
}
```

2.2 DLL bouwen (Intel C)

Gebruik een Intel C/C++ command prompt:

```
icl /LD rng.c /Fe:rng.dll
```

Resultaat: rng.dll met export RNG_NEXT.

3. Rocket COBOL 11.0: aanroep van de RNG DLL

De COBOL-kant gebruikt een COMP-2 variabele (8-byte IEEE double), die by reference wordt doorgegeven.

3.1 COBOL-programma

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. TEST-RNG.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 RNG-VALUE COMP-2.  
  
PROCEDURE DIVISION.  
CALL "RNG_NEXT" USING RNG-VALUE  
END-CALL  
  
DISPLAY "Random value = " RNG-VALUE  
  
STOP RUN.
```

3.2 Waarom dit werkt

- C: void __stdcall RNG_NEXT(double *value)
 - COBOL: COMP-2 → 8-byte double
 - Beide talen gebruiken pass-by-reference
 - Exportnaam is exact RNG_NEXT
-

4. Variant: integer RNG (1..N)

4.1 C-broncode (rng_int.c)

```
#include <stdlib.h>
#include <time.h>

static int rng_initialized = 0;

static void rng_init(void) {
    if (!rng_initialized) {
        srand((unsigned int)time(NULL));
        rng_initialized = 1;
    }
}

__declspec(dllexport) void __stdcall RNG_NEXT_INT(int *maxval, int *result) {
    int m;
    double r;

    rng_init();
    m = (*maxval <= 0) ? 1 : *maxval;

    r = (double)rand() / (double)RAND_MAX;
    *result = (int)(r * m) + 1;
}
```

4.2 COBOL-aanroep

```
01 MAXVAL PIC S9(9) COMP VALUE 100.
01 RNG-INT PIC S9(9) COMP.

CALL "RNG_NEXT_INT" USING MAXVAL RNG-INT
END-CALL

DISPLAY "Random int = " RNG-INT.
```

5. Conclusie

Met Intel C kun je eenvoudig een Windows DLL bouwen die random getallen genereert en volledig compatibel is met Rocket COBOL 11.0. Door gebruik te maken van `__stdcall`, voorspelbare exportnamen en eenvoudige pointer-interfaces, ontstaat een robuuste interoperabiliteitslaag tussen beide talen.