

# Volledige Ada Integratiehandleiding

## Volledige Ada Integratiehandleiding

### 1. Doel en uitgangspunten

Deze handleiding beschrijft hoe je **GNAT Ada 2021** veilig en schoon integreert in een bestaande native toolchain met:

- Visual Studio 2022 (MSVC toolset)
- Intel C
- IFX Fortran
- Rocket COBOL 11.0
- ML64-assembly

De integratie is **stand-alone**, zonder IDE-plugins en zonder PATH-vervuiling.

---

### 2. Installatie van GNAT Ada 2021 (geïsoleerd)

1. Installeer GNAT in een eigen prefix, bijvoorbeeld:

```
C:\Lang\GNAT2021\
```

2. Schakel alle opties uit die PATH of file-associations wijzigen.
3. Maak een GNAT-omgeving:

```
@echo off
set GNAT_HOME=C:\Lang\GNAT2021
set PATH=%GNAT_HOME%\bin;%PATH%
```

4. Gebruik GNAT uitsluitend via deze batchfile.
- 

### 3. Projectstructuur

```
D:\Projects\Interop\
ada\
src\
obj\
lib\
build_ada.bat
c\
fortran\
cobol\
bin\
```

---

## 4. Ada ↔ C Interoperabiliteit

### 4.1 Ada-export

```
package Math_Ops is
function Add_Ints (A, B : Interfaces.C.int) return Interfaces.C.int;
pragma Export (C, Add_Ints, "ada_add_ints");
end Math_Ops;
```

### 4.2 C-header

```
int ada_add_ints(int a, int b);
```

---

## 5. Ada Buildscript

### 5.1 Objecten genereren

```
@echo off
set ADA_SRC=D:\Projects\Interop\ada\src
set ADA_OBJ=D:\Projects\Interop\ada\obj
if not exist "%ADA_OBJ%" mkdir "%ADA_OBJ%"
pushd "%ADA_SRC%"
for %%F in (*.adb) do gnatmake -c -g -O2 -D"%ADA_OBJ%" %%F
popd
```

### 5.2 Static library

```
ar rcs D:\Projects\Interop\ada\lib\libada_bridge.a D:\Projects\Interop\ada\obj\*.o
```

---

## 6. Linken met MSVC / Intel / Fortran

### 6.1 Directe MSVC-link

- Voeg ada\obj toe aan *Additional Library Directories*.
- Voeg GNAT runtime libraries toe (verkregen via gnatmake -v).

### 6.2 Ada als DLL

- Gebruik een .gpr project met Library\_Kind => "dynamic".
  - Exporteer functies via pragma Export (C, ...).
  - Importeer in C/Fortran via \_\_declspec(dllimport).
-

## 7. Initialisatie vanuit C/Fortran/COBOL

```
extern void adainit(void);
extern void adafinal(void);
int main() {
    adainit();
    int r = ada_add_ints(2, 3);
    adafinal();
}
```

---

## 8. Debugging

### Visual Studio

- Mixed-mode debugging werkt.
- Ada-modules verschijnen als native modules.

### GDB

```
gdb bin\your_program.exe
```

- Breakpoints in Ada-bron zijn mogelijk.
- 

## 9. Veelvoorkomende valkuilen

- **Geen GNAT in PATH** zetten.
  - **C-ABI strikt volgen** met Interfaces.C.\*.
  - **GNAT runtime libraries** niet vergeten bij MSVC-link.
  - **adainit/adafinal** verplicht bij niet-Ada main.
- 

## 10. End-to-end Pipeline

1. Ada code schrijven.
  2. Build via build\_ada.bat.
  3. C-header opnemen in C/Fortran/COBOL.
  4. Objecten of library linken in VS.
  5. Debuggen via VS + GDB.
-

## 11. Conclusie

Deze handleiding levert een volledige, schone en stabiele integratie van GNAT Ada 2021 in een moderne multi-language native toolchain zonder IDE-vervuiling. Je behoudt volledige controle over ABI, build-stappen en debugging.