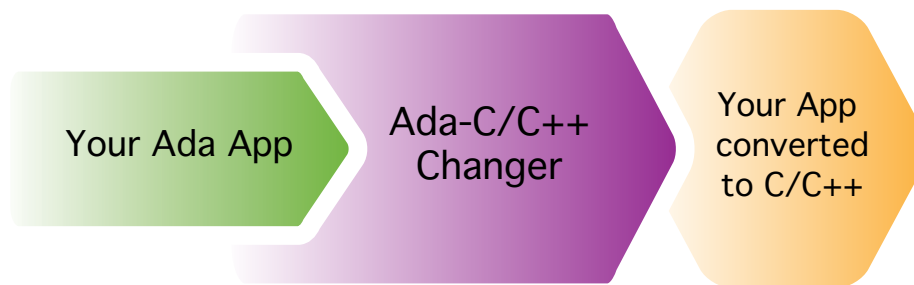# Ada-C/C++ CHANGER®

AUTOMATICALLY CONVERT Ada CODE TO C/C++

Its becoming much harder to find Ada programmers and Ada tools supporting your new hardware. The available Ada tools in the market are either unsupported or very expensive, unlike the equivalent C/C++ tools. The Ada-C/C++ Changer tool automatically converts Ada code to C/C++, without the need of having to manually re-write. This conversion process eliminates the need for a costly and tedious code re-write and provides developers extensive cost and time savings. The resultant outputted code is efficient and readable C/C++, exactly matching the semantics of the original Ada program. Optionally, the resultant C/C++ code can also be integrated with the OS Abstractor platform to obtain real-time performance, portability and stability across a multitude of OS platforms, thereby greatly minimizing any needed re-hosting effort now or in future.

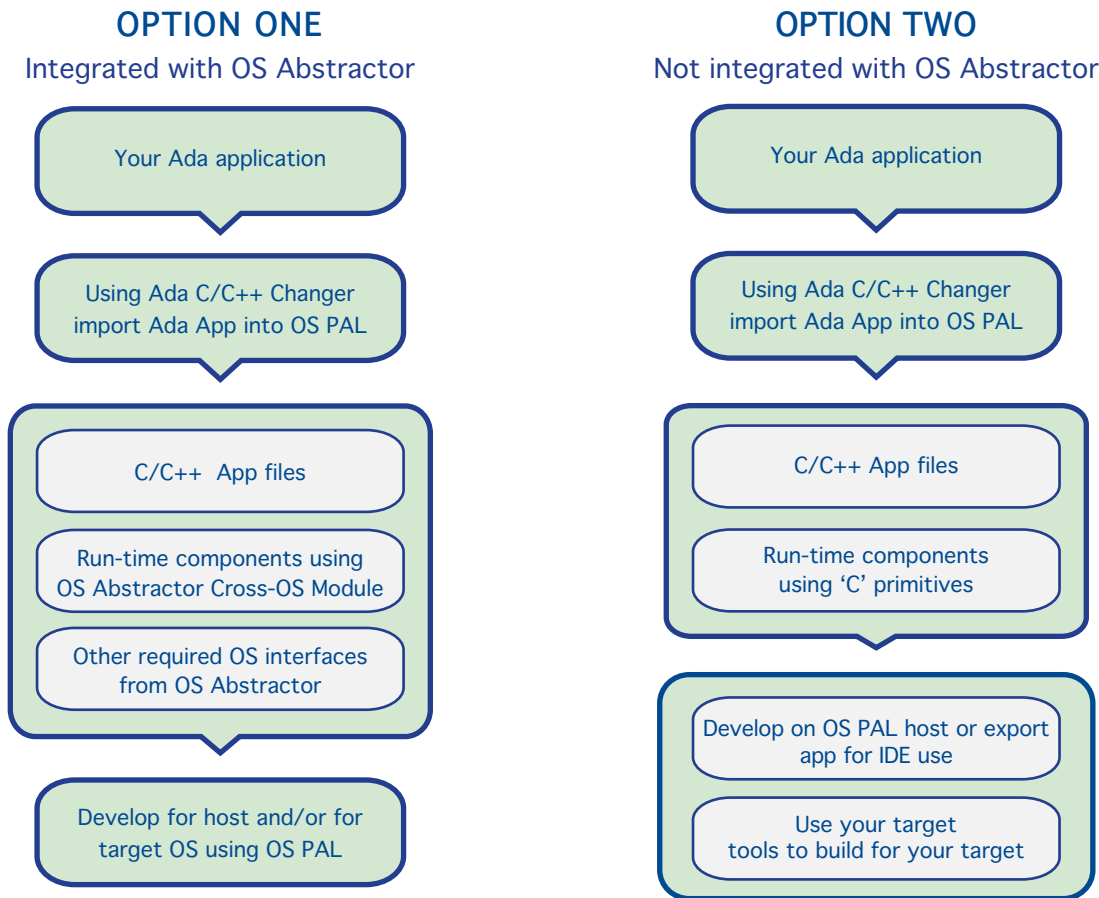Your Ada App → Ada-C/C++ Changer → Your App converted to C/C++

Ada-C/C++ Changer is a component of OS PAL, a robust porting and abstraction solution platform based on the Eclipse framework and GNU tools. Ada-C/C++ Changer reads the source code directly from your Ada source directories and then compiles them to generate equivalent C/C++ code that is fully readable and maintainable.

Below are the Target Operating Systems supported by Ada-C/C++ Changer:

Android®	MQX®	Solaris®	VxWorks® 6x/5x
Linux® 2.4/2.6	NetBSD®	ThreadX®	Windows® XP/Vista/7/CE
LynxOS®	Nucleus®	T-Kernel®	
LynxOS-SE®	QNX® Neutrino® RTOS	µC/OS III®	
µITRON®	RT Linux®	Unix®	

If your Ada application requires any additional 'C' binding, they can be easily added to the generated C/C++ output and compiled using either the supplied GNU compiler or any other preferred C/C++ cross-compiler to generate the application executable for your target. Ada-C/C++ Changer optionally includes Ada-PAL Compiler to allow developers to convert only a portion Ada code to C/C++, while keeping the rest in Ada. Optional integration with OS Abstractor allows the converted C/C++ code to utilize OS Abstractor's scheduler primitives and enable your converted code to run on a wide variety of target OS platforms.

The following diagrams depict two different ways that Ada-C/C++ Changer can be utilized during a conversion project:

OPTION ONE
Integrated with OS Abstractor

```
Your Ada application
        ↓
Using Ada C/C++ Changer
import Ada App into OS PAL
        ↓
┌─────────────────────────────────┐
│  C/C++  App files               │
│                                 │
│  Run-time components using      │
│  OS Abstractor Cross-OS Module  │
│                                 │
│  Other required OS interfaces   │
│  from OS Abstractor             │
└─────────────────────────────────┘
        ↓
Develop for host and/or for
target OS using OS PAL
```

OPTION TWO
Not integrated with OS Abstractor

```
Your Ada application
        ↓
Using Ada C/C++ Changer
import Ada App into OS PAL
        ↓
┌─────────────────────────────────┐
│  C/C++ App files                │
│                                 │
│  Run-time components            │
│  using 'C' primitives           │
└─────────────────────────────────┘

┌─────────────────────────────────┐
│  Develop on OS PAL host or export│
│  app for IDE use                │
│                                 │
│  Use your target               │
│  tools to build for your target │
└─────────────────────────────────┘
```

Noteworthy Features:

- Supports conversion Ada 95 and parts of Ada 2005 source code to ANSI C/C++ output
- Using Ada-C/C++ changer does not require you to convert all of the Ada Code in your application
  - > The rest of the Ada code can be compiled in using the Ada-PAL Compiler option
- Includes GNAT Ada compatibility option
- Includes POSIX Ada interface package to assist in transitioning code that relies on POSIX features
- Preserves Ada code's comments, files structures and variable names to ease with ongoing code maintenance
- Keeps "static" trees as symbolic expressions rather than evaluating them to literal values
- Preserves original Ada global/local variable names to a great extent in the generated C/C++ code
- Ada language defined exceptions can be made available in the converted C/C++ code
- Provides integration of Ada and C/C++ projects using OS PAL and allows host platform app development
- Provided various user configurable options to control the format and style of the C/C++ output

MAPUSOFT

www.mapusoft.com
1.877.MAPUSOFT