

FORTRAN to C/C++ conversion service

Version 1.0

19 September 2002

Stromasys

P.O.Box 156

1228 Plan-les-Ouates

Switzerland

Tel: +41 22 794 1070

Fax: +41 22 794 1073

www.stromasys.com

This document is provided for information and is not a legally binding offer. Stromasys reserves the right to change service specifications without prior notice or retire the specific service offered. Trademarks are the property of their respective owners.

Service availability

The FORTRAN to C or C++ programming language conversion service is available as part of a platform migration project or as a separate service. In the latter case a typical project is split into a fixed-price project evaluation phase and an execution phase, based on an agreed cost per line.

The fixed price evaluation delivers a detailed project plan within four weeks of its commencement and includes a cost estimate for the conversion. If the project is ordered within two months of completion of the project plan, this estimate can be considered as a fixed-price quotation.

The FORTRAN to C and C++ programming language conversion service provided by Stromasys is a risk free way to transform older FORTRAN applications into modern C or C++ code, while maintaining the original application functionality. Such conversion can generate substantial cost savings; older tools, development platforms and maintenance skills are no longer required. Modern platforms and programming languages make it easier to attract good software engineers, maintain code and offer a wider choice of powerful development platforms. Programming language conversion can significantly ease the transition to a new system environment.

Stromasys provides a comprehensive service designed to limit the involvement of your own staff to the final acceptance testing after delivery of the translated applications. We can tailor the conversion process to company specific coding standards, and replace middleware and system calls. Several levels of refinement are available, ranging from a rapid conversion for unmodified further use to development of customer specific methods and classes. In all cases, the original functionality is maintained by default.

Service components

The FORTRAN to C/C++ programming language conversion service is based on the following main components:

- A set of source code analysis tools to gather statistical information and to detect constructions and extensions that cannot be converted automatically. This information is used to tune the F2C/CPP converter and to extend/develop the C/C++ runtime libraries if necessary. This analysis also covers the system functions of host platform operating system, an essential part to be considered in the migration process.
- The F2C/CPP converter engine to handle automated volume conversion of the FORTRAN source code. The F2C/CPP converter is parameter controlled and manually adjusted based with the information obtained in the first step.
- A C/C++ run-time library to support the translated FORTRAN code
- Specific extensions to efficiently handle RSX/DEC FORTRAN.
- Specific emulation libraries for Operating System dependent code segments. Stromasys developed technology templates for migrations between different host platforms (e.g. from RSX11 and VAX VMS/OpenVMS to industry-standard UNIX or Windows 2000 systems).

The conversion service includes

- A feasibility analysis to determine manual and automatic components.
- Pre-conversion code preparation, cleaning up, filtering and processing with 'standardization' tools.
- The analysis of OS-specific extensions and packages used in the application.
- Bulk conversion of FORTRAN code to C or C++ using the F2C/CPP converter.
- Generation of OS-specific migration components for the new platform.
- Post-conversion clean-up (e.g. inline comments)
- Manual conversion of the outstanding conversion issues as reported by the initial analysis and F2C/CPP.
- Converted source code style improvements and redesign (if required).

Code conversion principles

From a design point of view, FORTRAN and C++ have radically different philosophies in implementing an application design. FORTRAN places an emphasis on routines and structural design, while C++ places its emphasis on classes and object orientation.

There are also some specific data types and some control statements that cannot be directly converted to C/C++ statements without introducing emulation. Most of the control statements and data types naturally convert to C++ with no loss in readability, but some compromises for FORTRAN-specific constructions are made in the converted code.

Conversion v.s. emulation

A predefined FORTRAN language environment consists of a set of standard functions provided by every FORTRAN implementation.

Some of the procedures implemented in these packages can be converted to native C++ functions, while others need special emulation provided by the FORTRAN support C++ runtime library.

RSX/DEC FORTRAN

The F2C/C++ translator design has a tunable open interface to adjust the conversion process to specific FORTRAN dialects. For instance, FORTRAN 77 on both VMS/OpenVMS (DEC FORTRAN 77) and RSX11 (RSX FORTRAN 77).

Code conversion principles

In comparison with GNU F2C and other available on the market FORTRAN-to-C conversion tools STROMASYS FORTRAN to C/C++ translator produces more readable code, so the need of manual editing is very limited. Most of FORTRAN operators and functions are converted to standard C/C++ language equivalents and only a small number are substituted from the supporting library (for real, complex arithmetic, character functions, input/output processing functions, etc.):

- Most of IO functions are converted to standard C library
- INTRINSIC functions are changed from the external library to C code
- Mathematical functions are using C library functions
- Multidimensional arrays are implemented
- All constant, defined with PARAMETER are #define
- Strings are presented as C-style NULL-terminated strings
- String operations are implemented via C library functions
- Variable initialization in type declaration is allowed
- Statement functions are converted to macro directives

Advantages of C++ usage comparing to pure C

Both FORTRAN to C and FORTRAN to C++ translators both are encapsulated in one application, which is named as STROMASYS F2C/CPP. Comparing to C, the conversion to C++ produces a code that from the reader's point of view is closer to the original FORTRAN source code style. This is achieved by using additional classes with features such as operator overloading, templates etc.

The most significant advantages of C++ conversion are as follows:

- Statements using complex arithmetic are practically identical to FORTRAN. Complex arithmetic is included in the DEC C++ class library.
- Multidimensional arrays with non-zero low bounds are supported in direct form. In C, single-dimensional arrays with computed indexes are used.
- String operations are supported like in FORTRAN.
- Function argument lists are implemented using variable references, like in FORTRAN.
- Relational expressions: using operation overloading the resulting code will resemble the FORTRAN source code, regardless of types used in expression.

FORTRAN conversion complexities

Several FORTRAN architecture essentials which don't have a direct equivalent in C/C++ are addressed in the FORTRAN to C and C++ translator as follows:

- COMMON statements are translated to C structures, macro directives are supplied to simplify access to common block members.
- EQUIVALENCE statements are translated into an array of size of the largest entity, making a #defines to produce aliases for entities and then to an element which looks like an ordinary statement.
- Conversion of input/output statements is mainly handled automatically.