• A Companion Processor is a mechanism for defining procedures and using data. It is most often a C compiler. The provisions for interoperation are described in terms of C.

• Interoperability is the property of being usable with both the Fortran processor and a companion processor.

• A Binding Label is a global identifier which bridges the gap in syntax between Fortran names and C identifiers. A Fortran procedure, module variable, or common block can be given a binding label, possibly different from its Fortran name.

• The BIND Attribute is used to specify interoperability of a derived type, procedure, procedure interface, or common block. It is also used to specify a binding label for a procedure, module variable, or common block.

Interoperability allows processing to be done using a mixture of languages. Fortran supports the following aspects of interoperability with C:

1. Procedures. A Fortran program can incorporate functions written in C. The Fortran code can invoke those C functions; conversely, the C functions can invoke Fortran procedures.

2. Data. The Fortran and C portions of a Fortran program can communicate data by argument passing or global data.

3. Files. Separate Fortran and C programs can communicate by means of files.

This chapter covers interoperability of procedures and data in a program that has a mixture of Fortran and C code. Interoperability of files is facilitated by stream access, discussion of which is integrated into the chapters on input/output.

A Fortran and a C entity are interoperable with each other if they have corresponding properties. The details of how their properties must correspond depend on whether the entities are types, data objects, or procedures. In some contexts, the interoperability of an entity in one language is discussed, regardless of whether there is a corresponding entity that it is interoperable with. An entity is interoperable in this sense if it has the properties necessary for interoperation.

The BIND attribute has multiple roles. For a derived type, the BIND attribute specifies interoperability. For a procedure, procedure interface, or common block, the BIND attribute specifies interoperability and a binding label. For a variable, the BIND attribute specifies a binding label; a variable without the BIND attribute may be interoperable, but it does not have a binding label.
15.1 Companion Processors

The interoperability features of Fortran are designed around interoperaction with C, but can indirectly support other languages as well. Each processor has one or more companion processors with which it can interoperate. The particular set of companion processors and the means of selecting among them are processor-dependent.

In order to be constructively useful, a companion processor must be capable of working with data and procedures that can be described in terms of C. The obvious case of a candidate companion processor is a C compiler. Compilers for other languages can also be companion processors if they are capable of interoperating with C. It might be said that C serves as the "lingua Franca" of programming languages.

For example, C++ compilers can use the "extern C" attribute to interoperate with C functions. Therefore, interoperaction between Fortran and C++ can be achieved through the common ground of C. Likewise, Ada compilers can interoperate with C and thus could conceivably serve as companion processors. One simple case that is easy to overlook is that the companion processor could be a Fortran compiler. The interoperability facilities of the language could be used to facilitate interoperaction of multiple Fortran compilers.

As a trivial case, a Fortran compiler could serve as its own companion processor; this trivial case allows a vendor to meet the requirement of supporting one or more companion processors even in an environment where there might be no other processors. Although the interoperability features have little direct relevance to such an environment, their support facilitates portability of code that might be used in other environments.

15.2 Binding Labels

Fortran and C have different rules on the validity and uniqueness of identifiers. In particular, C identifiers are case sensitive, but Fortran identifiers are not. The concept of a binding label bridges this gap and provides a means for Fortran code to refer to C identifiers. A binding label follows the rules for C identifiers.

A binding label is the global identifier by which a Fortran variable, common block, or procedure is known to the C compiler. If an entity has a binding label, it can be referred to by that identifier in C code. A variable or procedure that has no binding label has no global identifier in C, but can still be interoperable and used in ways that do not require a global identifier. For example, a global identifier is not needed for a procedure argument.

The BIND attribute specification for a variable, common block, procedure, or procedure interface confers a binding label, either explicitly or implicitly. If the BIND attribute specification includes a NAME specifier, then the binding label is the character value so specified, with any leading or trailing blanks removed; the case of the letters is significant in a binding label specified this way. If there is no NAME specifier, the binding label is the name of the variable, common block, procedure, or procedure interface, with all letters in lower case.