

# Fortran 77 Extension Builder for Python

Luke Bishop

*Photon Research Associates, Inc.*

## Abstract

The use of Fortran codes in scientific, research and development environments, such computational chemistry, radiation transport, and modeling and simulation is well established. A flexible paradigm of code development and scientific research is possible making use of interactive, scripting languages controlling and "steering" computational-intensive codes developed in traditional, compiled languages. A general-purpose tool (Extension Builder) has been developed to build Fortran 77 extension modules and to define public interfaces exposing Fortran 77 variables and functions.

## Extension Builder

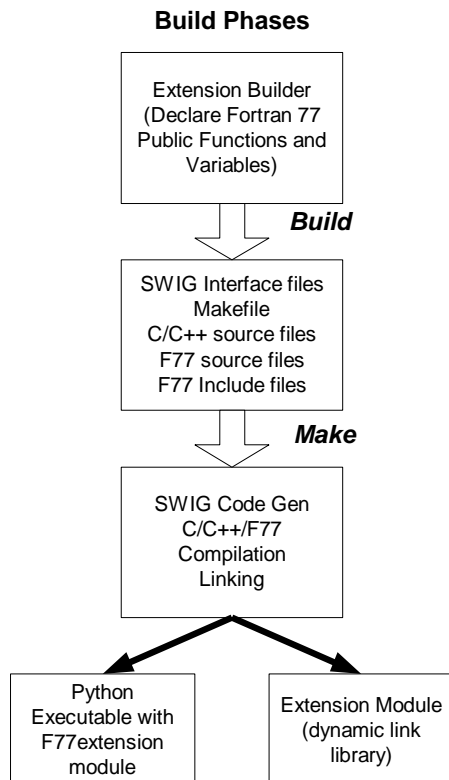
Fortran 77 developers and researchers use Extension Builder to declare a public interface to Fortran 77 procedures and variables. Variables of all common Fortran 77 data types (up to 2 dimensions) can be declared using Extension Builder. Function prototype definitions describe the Fortran procedures to be made public. Function prototype definitions can include parameters of all common types. Extension

Builder was developed in C++ using a cross-platform graphical user interface tool. Currently, Extension Builder generates MAKE files compatible with Silicon Graphics systems (IRIX 6.4 and later) and Win32 Microsoft Visual Studio IDE (MSVC5.0 and Digital Visual Fortran).

## Building Extensions

Developers specify function prototypes for Fortran 77 procedures to be made public to Python users. Function parameters can be declared with "input", "output", or "inout" access, as defined in Extension Builder's SWIG (Simplified Wrapper and Interface Generator) typemaps. Likewise, Fortran variables can be declared with default values with minimum and maximum range values. Based on declarations, a code generator creates wrapper files, a SWIG interface file, and a MAKE file. The MAKE file can be executed from within Extension builder, from within an IDE, or stand-alone. A target dynamic link library or static python executable (Unix only) is then built.

Extension Builder utilizes SWIG extensively to build the wrapper interfaces. Customized SWIG typemaps are used to map standard data types and Numeric Python (NumPy) array types from the python user environment to the Fortran extension library via the public functions and variables.



## Fortran 77 Variable and Function Mapping

