

Xcf77: An OSF/Motif based GUI for Cf77

Joachim Backes, University of Kaiserslautern,
Computer Center [RHRK], Kaiserslautern, Germany

ABSTRACT: *xcf77 is a X-Window and OSF/Motif based access to the Cray cf77 Fortran programming environment, executing locally under UNICOS or on arbitrary workstations. It provides a menu controlled call of F77 compiler, gpp, fpp, fmp and (restricted) segldr. All cf77, cft77, fmp, fpp and gpp options are supported. Invalid or unreasonable combinations are rejected before the compiler phases are executed. Xcf77 can produce makefiles by scanning all involved sources. It offers interfaces for source editing and printing. Optionally, the actions performed by cf77 can be stored in history files and reexecuted later on. A similar interface has been developed for SGI Fortran77.*

xcf77 has its origins in a tool developed for the VP supercomputer series.

1. Introduction

Today's workstations and X-terminals offer a lot of comfort by the use of graphical interfaces. But UNIX itself –with its shells, compilers, utilities...– has cryptic or primitive interfaces and, additionally, restricted or missing help features. This is an extremely unsatisfactory situation, especially for beginners. The same problem arises for people which are working in a well-outlined project in a rather restricted environment, but without any necessity for a UNIX training or course.

These problems are multiplied, for example, if the UNIX machine runs in a backend environment, accessible by one or more heterogeneous operating systems.

A pregnant example for this lack is the *compile/link* environment of the Cray Fortran compiler cf77: the *Ready Reference* itself consists of **42 pages** presenting the compiler and phases options, and this without any detailed help!

UNICOS products such as *xsam* or other new *x-utilities*, which came up in the last years, are only first approaches to a well designed GUI representation. But they are no definite and no sufficient solutions.

What is missing, especially for *fortran*- or *c*-developers, are simple to learn but powerful and easily remembered graphical interfaces, including inconsistency control, help features and so on.

Xcf77 is an attempt to implement such a well understandable, self explaining, robust and portable interface to the Cray Fortran77 development environment, in accordance with the OSF/Motif Style Guide. And: *xcf77* tries to avoid the disadvantages and problems reported above.

2. Design

2.1 Structure

According to the wide spread requirements for such a tool, the functions of *xcf77* are arranged in groups, visible to the user, as shown in the following figure:

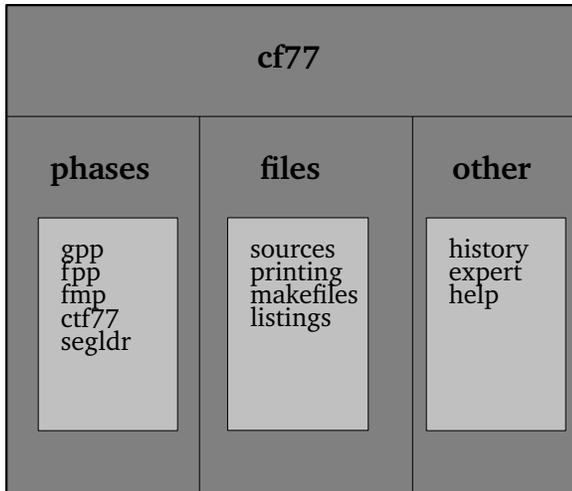


Figure 1. Xcf77 logics

From the user's point of view, the functions listed above are realized by separate Motif windows on the screen, logically grouped in the following manner:

- Compiler control; manages
 - xcf77 itself,
 - gpp,
 - fpp,
 - fmp,
 - ctf77,
 - segldr.
- File control; manages
 - Source editing,
 - Compiler listing browser,
 - Makefile generation,
 - Printer services.
- Other features:
 - Local and remote execution control,
 - Exit control,
 - History management,
 - A direct command mode for experienced users.
- Online help, directly available by the F1 key, based on the *man(1)* pages of xcf77 and the subordinated tools.

2.2 Limitations

The implementation of xcf77 is based on *X-Windows* as window system and *OSF/Motif* as toolkit interface. Additionally, it is supported by an *interface builder*,

managing and generating the GUI part (not the internal implementation part). As an X-program, xcf77 supports client/server architectures. Xcf77 is portable between UNICOS and standard workstations

Internal, but not external disadvantages are the complexity of the X-Window and Motif programming and the awkward debugging of the "look and feel" of such an application.

And so what to do if the Motif libraries are not available on UNICOS, but only on the workstations? This has an additional impact on the user interface; more exact, the support of a *distributed development environment* has to be guaranteed:

- Location of the source files,
- Compiler start and process management,
- Access to the compiler listings,
- Makefile handling.

To solve these problems, xcf77 has been designed for the use in *two modes*: running *natively* under UNICOS (using CVT, the Cray Visualization Tool) and, on the other hand, running on almost all workstation platforms (SGI, HP, RS6000, SUN, CS6400 ...). In the workstation case, all UNICOS activities are internally called by the *rsh(1)* command (not by remote procedure call). This needs user specific UNICOS *.rhosts* entries.

3. Xcf77 functions

The root window of xcf77 (as a graphical representation of xcf77) has full control over all phases and subtools. Appropriate to the requirements of the OSF/Motif style guide, xcf77 offers File services, compiler services, specials and help. These services are activated by menu bars, push and toggle buttons.

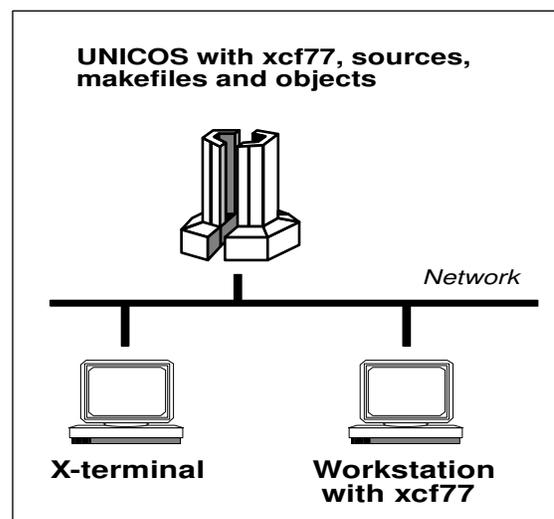


Figure 2. Xcf77 within a network

3.1 File services

By the file services, the user controls the following points:

- Decision whether xcf77 runs locally under UNICOS, or on some workstation. When running on workstations, the CRAY machine's internet address, the CRAY userid and the CRAY search path have to be specified. Reasonable default values are taken, they can be put into X resource files.
- Editor used for the source file editing session.
- Command used for printing.
- If the user wants to log the compile activities, he can save the corresponding cf77 commands in a history file, or he can load an arbitrary stored session for the reexecution of some commands.
- Xcf77 exit

3.2 Compiler phases control

The compiler phases are selected by toggle buttons. Pressing them will pop up the correspondend phase windows for

- gpp (General Preprocessor),
- fpp (Fortran dependency analyzer)
- fmp (Fortran multitasking translator),
- cft77 (The genuine compiler),
- segldr (Linkage editor),
- but no support for fort77!

All options of cf77 and the phases are supported (except segldr), and they are checked for consistency. The generation is done only on demand, and the options are collected together for the final use in the cf77 control window. Here the generated options can be edited, if necessary, so there is given a full and flexible control over the used options.

The fact that the cf77 functions on the one hand, and the functions of the phases on the other hand, are partially *overlapping*, produces some confusion for the users. This problem has been solved by offering the overlapping (possibly implemented with different names) options only in the cf77 control window, not in the window of that phase. This procedure guarantees orthogonality, which otherwise is missing.

Additionally, the most important parameters for Fortran developers are available in the CF77 window:

- Source files, include files and source file format control,
- Debug control.

- Optimization control,
- Selection of the phases to be executed (-Z option),
- Cflist control.

3.3 Makefile generation

After the option generation, the user can produce makefiles. In this case, all source files and include files (until to an adjustable depth, but not the system includes) are scanned for dependencies. These dependencies together with the generated compiler options and segldr options are collected in a makefile for the use by the make (or gmake) utility.

3.4 Compiler listing

When the user requires a compiler listing by specifying the corresponding cft77 options, a listing window will automatically appear, containing the source, error messages and other. This listing can be printed, and it can be searched for errors.

3.5 Expert mode

Experienced users can skip the xcf77 option generation. It is possible to run cf77 and the other compiler utilities in the normal UNIX line mode, but within xcf77. Especially, when the history file feature is used, recalled compiler runs are performed in this mode too. This, as a restriction, means that recalled compiler calls do not reproduce the graphical layout of the phase windows.

4. Acceptance

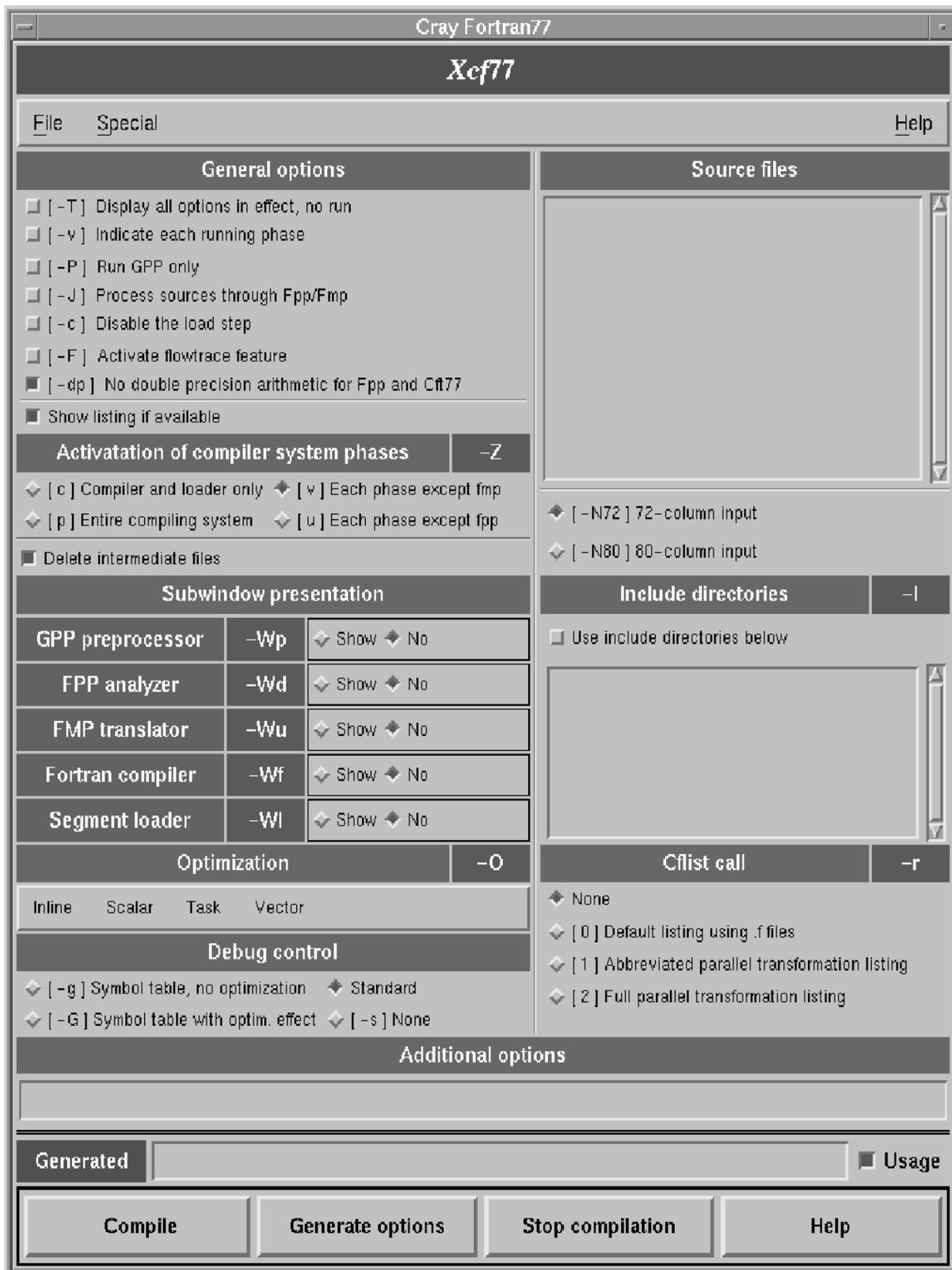
At our university computer center, the introducing UNICOS cf77 programming courses all are based on xcf77, using X-terminals. Xcf77 is well accepted and used by scientists and by students.

5. References

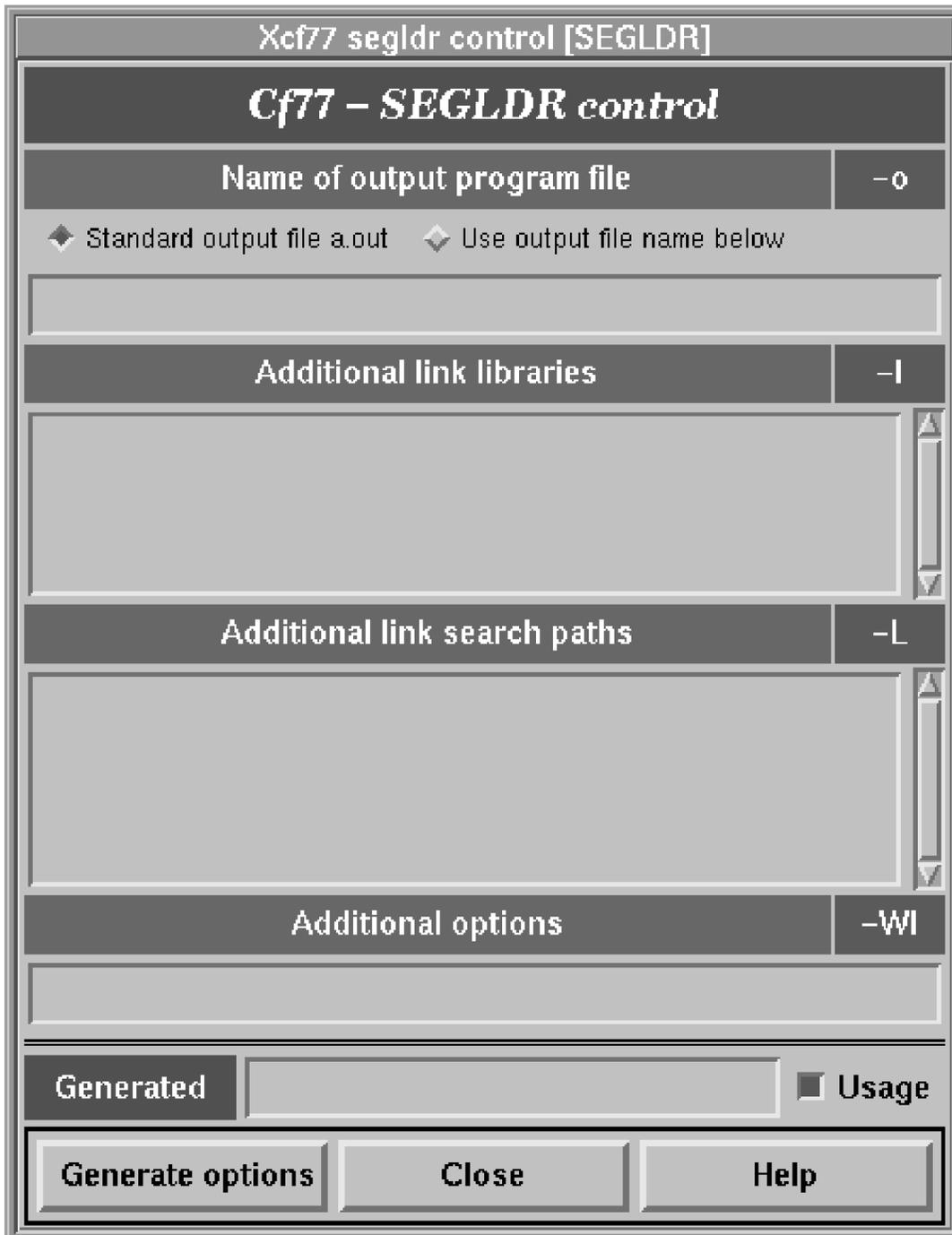
- 1) Cray Research, Inc., **CF77 Compiling System Ready Reference, SQ-3770 6.0, 1993**
- 2) Cray Research, Inc., **CF77 Fortran Language Server, Reference Manual, SR-3772 6.0, 1993**
- 3) O'Reilly & Associates, Inc., **The Definitive Guides to the X Window System, all volumes**
- 4) Douglas A. Young, **Programming and Applications with Xt, OSF/Motif Edition, Prentice Hall, 1990**

6. Layout of some xcf77 windows

6.1 Xcf77 control window layout



6.2 Cft77 control window



6.3 Xcf77 environment control

The image shows a dialog box titled "Xcf77 environment". It contains several sections for configuration:

- Xcf77 environment** (title bar)
- Xcf77 environment*** (decorative header)
- Cray site** (section header)
- Radio buttons for **Local** and **Remote**.
- Local (or remote) source file directory** (section header)
- An empty text input field.
- The remote CRAY's internet address** (section header)
- Text input field containing `ymp.rhrk.uni-kl.de`.
- Remote user name** (section header)
- Text input field containing `backes`.
- Remote search paths** (section header)
- Text input field containing `/usr/local/bin:/usr/rhrk:/usr/bin/X11`.
- Close** (button)