# Cross Reference Tool (XREF)

version 1.0

Peter Olin

1997-05-02

Typeset in LaTeX from SGML source using the DOCBUILDER 3.0 Document System.

# Contents


1 XREF User's Guide 1

1.1 The Cross Reference Tool . . . 2

Purpose . . . 2

Limitations . . . 2

Invocation . . . 2

The Main Window . . . 3

The Graph Window . . . 4

1.2 XREF Release Notes . . . 7

XREF 1.0.1 . . . 7

XREF 1.0 . . . 8

2 XREF Reference Manual 9

2.1 xref (Module) . . . 10

List of Figures 11

Module and Function Index 13

# Chapter 1

# XREF User's Guide

The Cross Reference Tool (*XREF*) performs statistical analysis on Erlang programs; providing the programmer with information about cross referencing between modules and functions etc.

# 1.1 The Cross Reference Tool

The Graphical Cross Reference Tool performs a static analysis of an Erlang program, resulting in information about:

> Note:
> This application will *not* be supported from next release of Erlang/OTP. Please, avoid to use it!

- inter-module dependencies
- undefined modules
- intra-module dependencies, i.e. between functions a module
- unused local functions
- unused exported functions
- inter-module recursion

## Purpose

The purpose of the Graphical Cross Reference Tool is to be of assistance during development and maintenance of Erlang software. It will help the developer to find undefined functions and to identify dependencies between modules and functions.

## Limitations

The analysis functionality cannot handle calls to `apply` and `spawn` where the applied or spawned function is a variable.

The analysis cannot handle higher order functions in general. So calls to funs and functions using for instance `lists:map` will not be handled properly.

The end result due to the limitations described will be that functions that are used in any of the ways described, but not defined, will not be listed as undefined, and any cross references due to such usage will not be visualized.

## Invocation

The Graphical Cross Reference Tool is started by typing the following in an Erlang shell:

```
xref:start().
```

# The Main Window

## Overview

The main parts of the Main Window are the *Files* list, the *Modules* list and the *Functions* list.

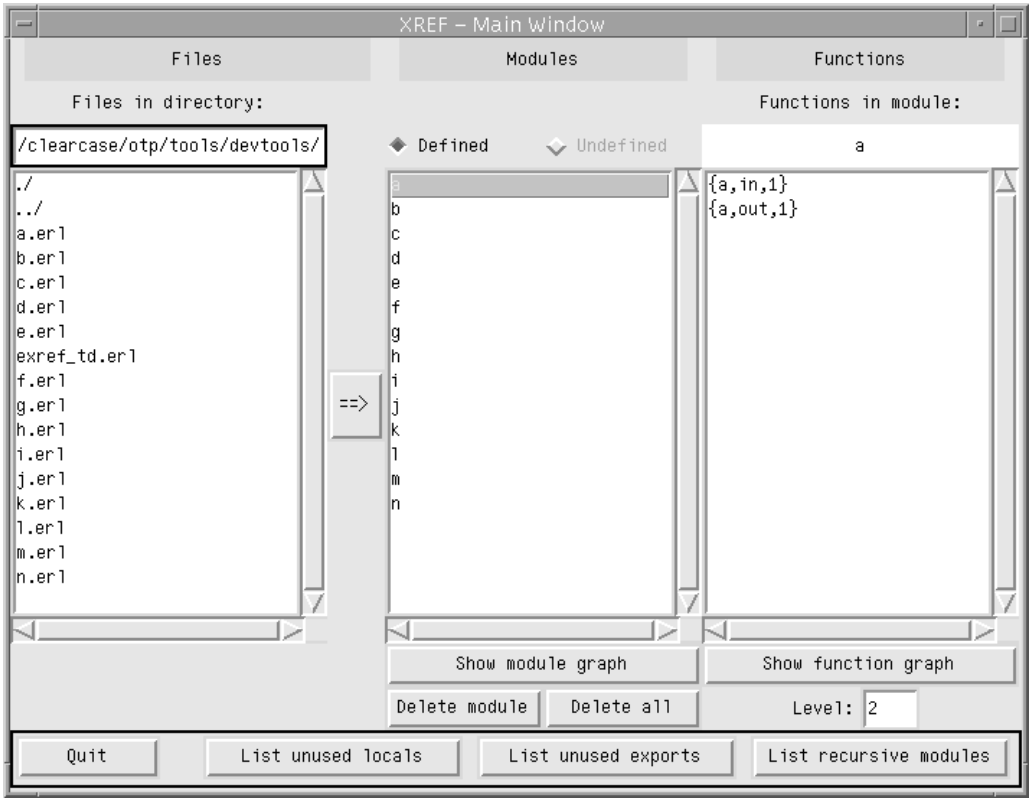


Figure 1.1: The Main Window showing a set of loaded modules.

## Files

### Moving in the directory structure

- A double click on a directory name in the *Files* list-box will open that directory.
- Place the mouse pointer in the entry field and type in the directory name, terminated with a (back)slash and press the `Enter` key.

**Loading files** To load one or more files, select a set of files and/or directories in the File list-box and then click on the right-arrow button placed to the right of the Files list.

If a directory is selected and loaded, all files with a `.erl` extension in the selected directory will be loaded, and subsequently appear in the Modules list. Since the files are analyzed as they are loaded this operation may take some time.

### Modules

The contents of the modules section is determined by the radio-buttons `Defined` and `Undefined`.

Depending on which of the radio-buttons is pushed, the contents of the module list will be as follows:

- If the `Defined` button is selected, all loaded modules plus any built in functions that are referenced by the loaded modules will be shown in the module list-box.
- If the `Undefined` button is selected, all modules that are referred to from other modules, but that are not currently loaded will be shown.

To view a graph of inter-module dependencies, click the `Show module graph` button. This will open a graph window showing the inter-module dependencies between the selected modules.

To remove a module from the set of analyzed modules, click the `Delete module` button. This will unload the selected modules.

To remove all modules from the analyzed set, click the `Delete all` button. This will unload all loaded modules

To view modules that are mutually dependent, click the `List recursive modules` button. The occurence of mutually dependent modules may indicate poor module design.

### Functions

When a function is analyzed, one can specify how many levels of function calls that should be traced from that function.

To view a function graph, select one or more functions in the function list-box, then click the `Show function graph` button.

To view all local functions that are not called, click the `List unused locals` button.

To view all exported functions that are not called, click the `List unused exports` button.

### Quitting

To quit the application click on the `quit` button.

## The Graph Window

### Overview

The graph window is invoked by clicking either of the two buttons `Show module graph` and `Show function graph`. It's purpose is to show a cross reference graph of selected modules/functions and their dependencies.

The nodes in the graph can be moved around manually. By using the functionality to relax a graph, all nodes that do not have a fixed position will be moved around automatically.

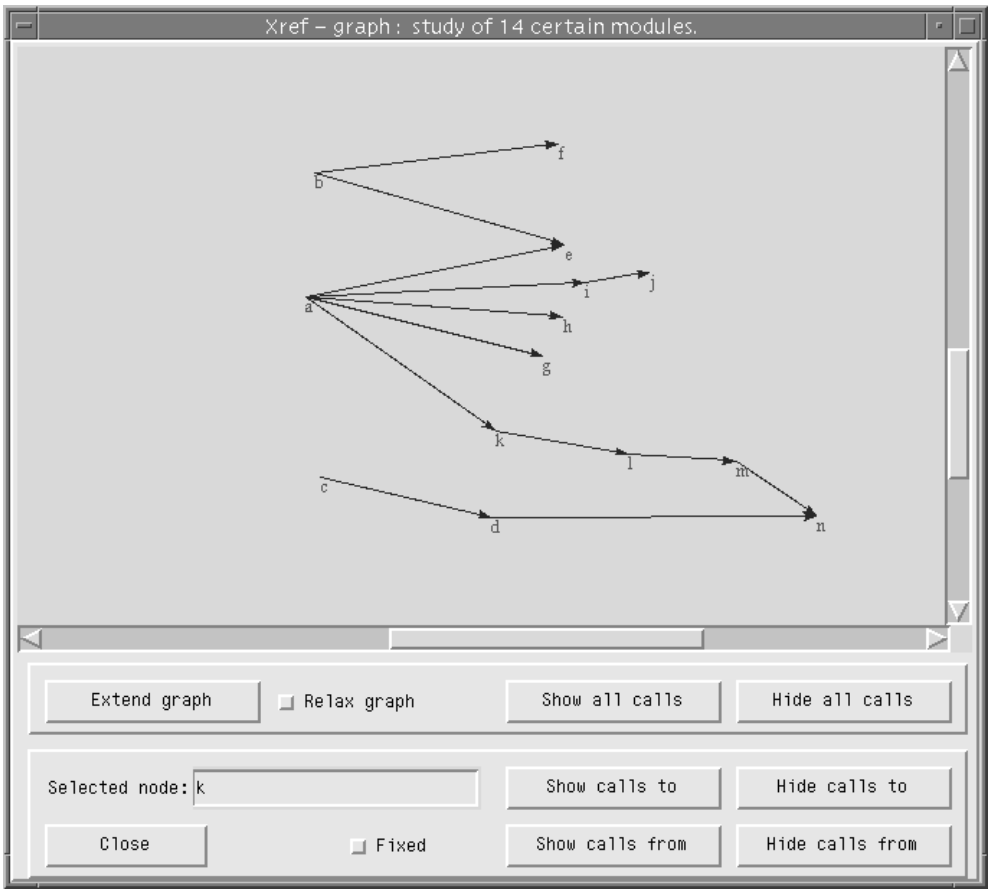


Figure 1.2: The Graph Window showing all references between a set of sample modules.

The functionality in the window is the same, independent of whether it was opened for modules or for functions, with the natural difference being that they show different objects.

The modules/functions are represented as nodes in a graph, and calls between them are represented as arrows. By default no edges are shown.

The modules/functions that were selected in the main window will be fixed nodes, i.e. nodes that will not be moved around by the relax function (see below). Fixed nodes are drawn in yellow.

Red nodes are non fixed nodes. And a green nodes represent a selected module/function..

## Graph operations

**Node manipulation** To select a node, click on it with the mouse pointer.

To deselect a node, click somewhere in the graph area, or on another node to select that node instead.

To fix the position of a node, select it and click the check-box button `Fixed`

**Showing cross references** To show cross references between modules/functions, press one of the `Show` buttons.

The buttons `Show calls to` and `Show calls from` operate on the selected node, whereas the `Show all calls` button will show all calls to and from all modules/functions in the set.

There is also a corresponding set of `Hide`-buttons, which hides the cross references in the same manner as described above.

**Graph layout** To automatically rearrange the nodes in the graph, click the `Relax graph` check-box button. To end the relaxation click once more one the same button.

This operation tries to minimize the edge crossings and with that making the graph more clear. It is a time consuming operation so the best thing to do is to having it run for a while, break and continue by hand.

## Closing the Graph Window

To close a Graph Window, click the `Close` button.

## See Also

There is also a brief reference page for the Graphical Cross Reference Tool in the module `xref`.

# 1.2 XREF Release Notes

This document describes the changes made to the graphical Cross Reference Tool.

## XREF 1.0.1

### Improvements and new features

Since the previous version some minor bugs and cosmetic changes have been made.

### Fixed bugs and malfunctions.

- The program no longer sets the current working directory.
- The problems that caused the program to occasionally crash at certain uses of the graphical interface has been fixed.
  Own Id: OTP-1382
- The problem in the `exref` module that caused analysis problem after *Delete Module* has been fixed.
  Own Id: OTP-1115
- A problem in `exref` has been fixed, so XREF no longer crashes on “List unused locals”
  Own Id: OTP-1973

### Incompatibilities with XREF 1.0

-

### Known bugs and problems

- There is no direct access to on-line help from the graphical cross reference tool.
  Own Id: OTP-1614
- The user interface is still non-standard and confusing.
  Own Id: OTP-1984
- Due to problems with `exref` Some important and informative output from the analysis is output in the shell, and not shown in the graphical user interface.This includes:
  - Warnings about occurences of `apply` with variable arguments.
  - Warnings about files of incorrect format.

  Own Id: OTP-1985

# XREF 1.0

## Improvements and new features

No new functions have been added.

Some bugs that caused the user interface to crash have been fixed.

Minor cosmetic changes to the user interface.

## Known bugs and problems

- The program sets the current working directory.
- The program will crash if the user navigates to a directory that contains a symbolic link that point to non-existent path name. Own Id: OTP-1382
- There is no access to on-line help.
- Due to a bug in the `exref` module the *Delete Module* function will cause some analysis functions to return incorrect information.
- The user interface is non-standard and confusing. Simple buttons are used when radio buttons should be used, buttons are used where menus should be used, buttons are not greyed out, different highlight colors are used, etc.
- Due to problems with `exref` Some important and informative output from the analysis is output in the shell, and not shown in the graphical user interface.This includes:
  - Warnings about occurences of `apply` with variable arguments.
  - Warnings about files of incorrect format.

# XREF Reference Manual

## Short Summaries

- Erlang Module xref [page 10] – Graphical Cross Reference Tool for Erlang Programs

## xref

The following functions are exported:

- `start() -> pid()`
  [page 10] Starts the Cross Reference Tool.

# xref (Module)

The Cross Reference Tool analyses an Erlang program statically and allows the user to view the results in a graphical format.

When started, the Cross Reference tool opens its main window which allows the user to specify which modules to include in the analysis.

From the main window the user may request analysis in different forms.

## Exports

```
start() -> pid()
```

Starts the Cross Reference Tool.

## See Also

Refer to the Tools chapter of the Erlang Development Environment User's Guide for a complete description of the Cross Reference Tool.

# List of Figures

Chapter 1: XREF User's Guide

1.1 The Main Window showing a set of loaded modules. . . . . 3

1.2 The Graph Window showing all references between a set of sample modules. . . . . 5

# Index

Modules are typed in *this way*.
Functions are typed in `this way`.

`start/0`
    *xref* , 10

*xref*
    `start/0`, 10