

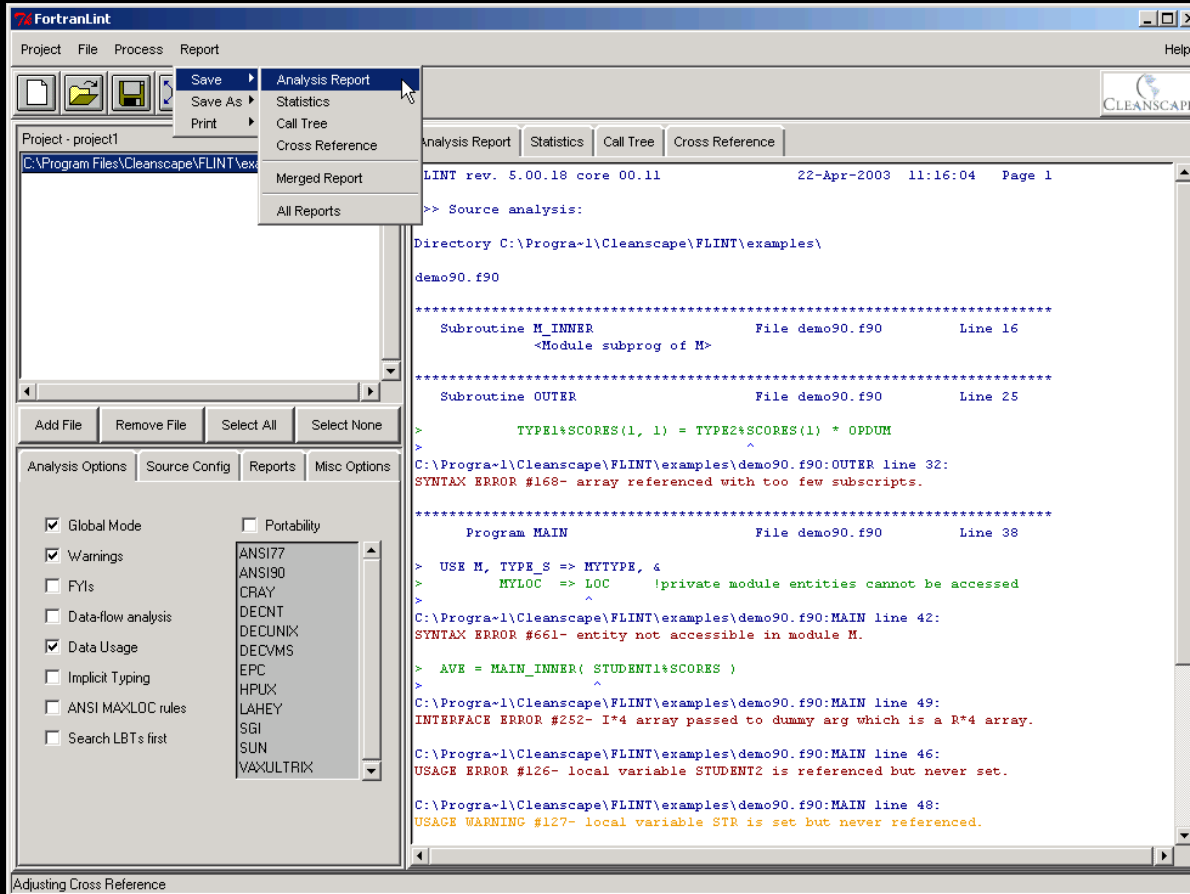
# How to stop Fortran programming problems at the source

*Cleanscape FortranLint Source Analyzer for Fortran*

*Tutorial and Demonstration*

**Keywords:** software programming tools, Fortran, static source code analyzers, c language, Fortran lint, Fortran source code analysis, Fortran language, software development automation

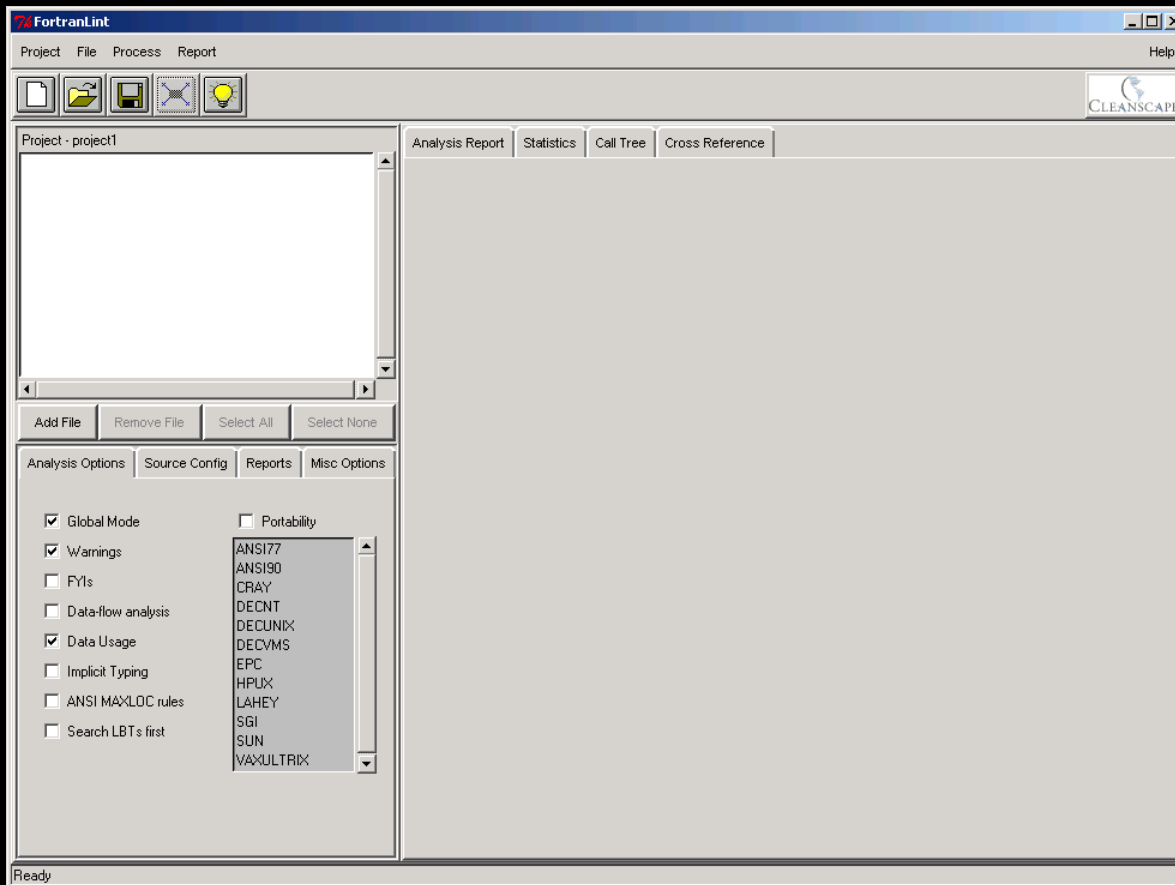




## Introduction

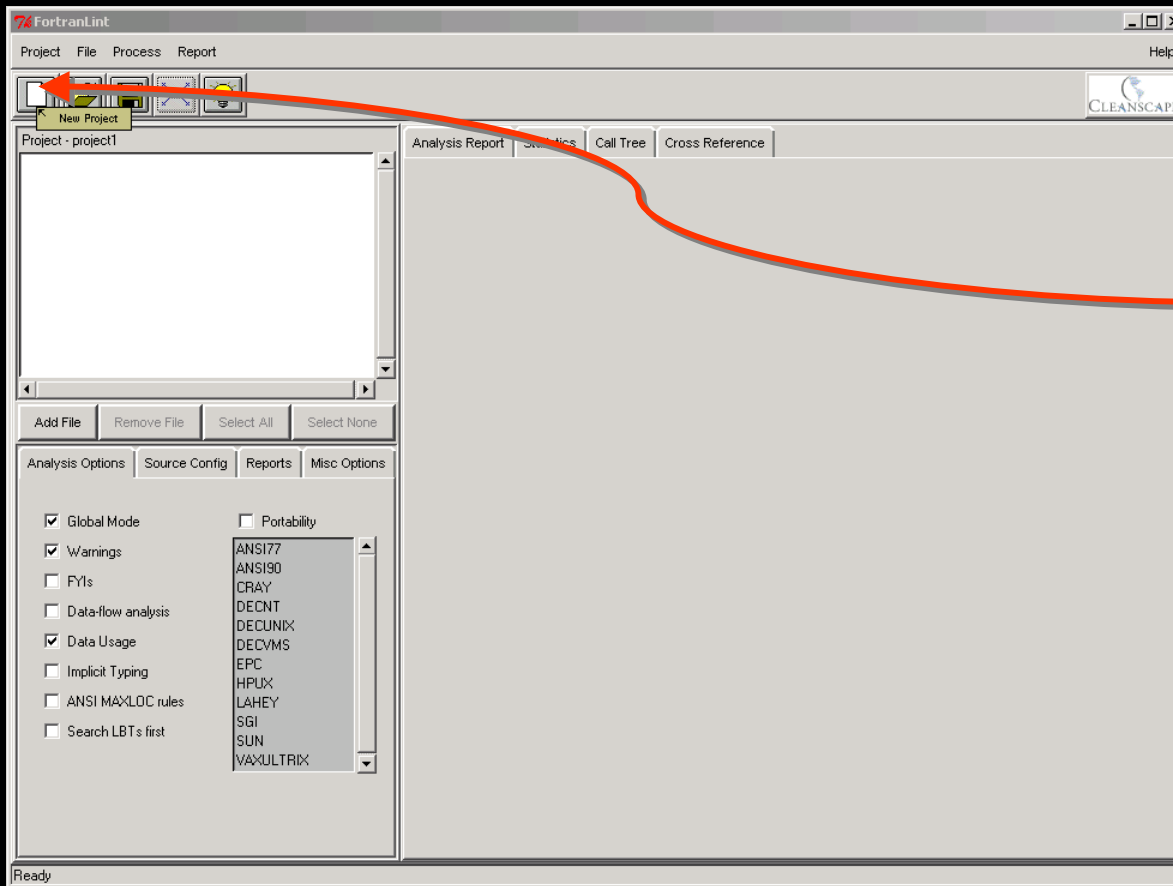
- ◆ Detect Fortran bugs and other problems that a compiler can't catch
- ◆ Conduct thorough assessment with global call interface checking, local dataflow analysis, and best practices information
- ◆ Conduct portability checks between multiple host environments an the ANSI F77/F90 standards
- ◆ Rapidly understand Fortran sources via call trees and cross references

# Graphical User Interface



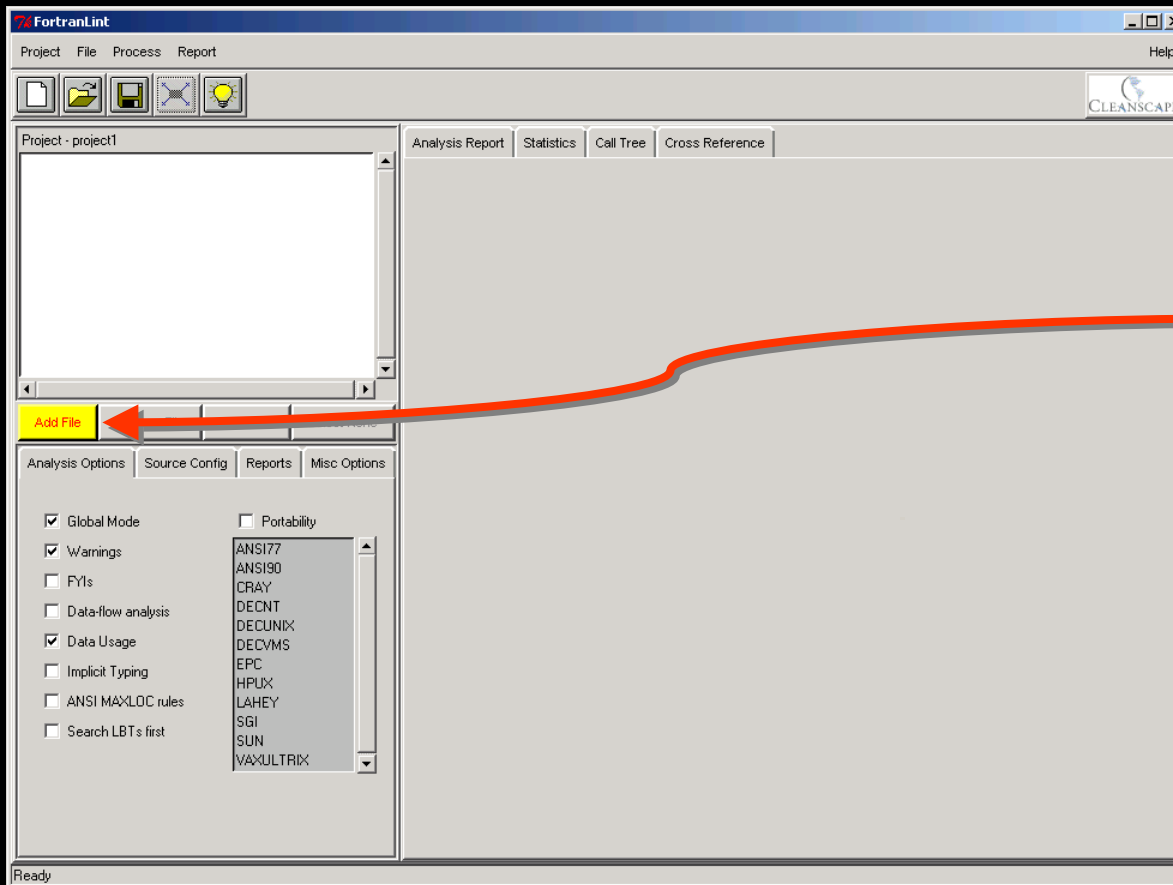
- ◆ Just click to access to powerful analysis features
- ◆ Simple to use
- ◆ No learning curve

# Start new project



◆ Select Project > New

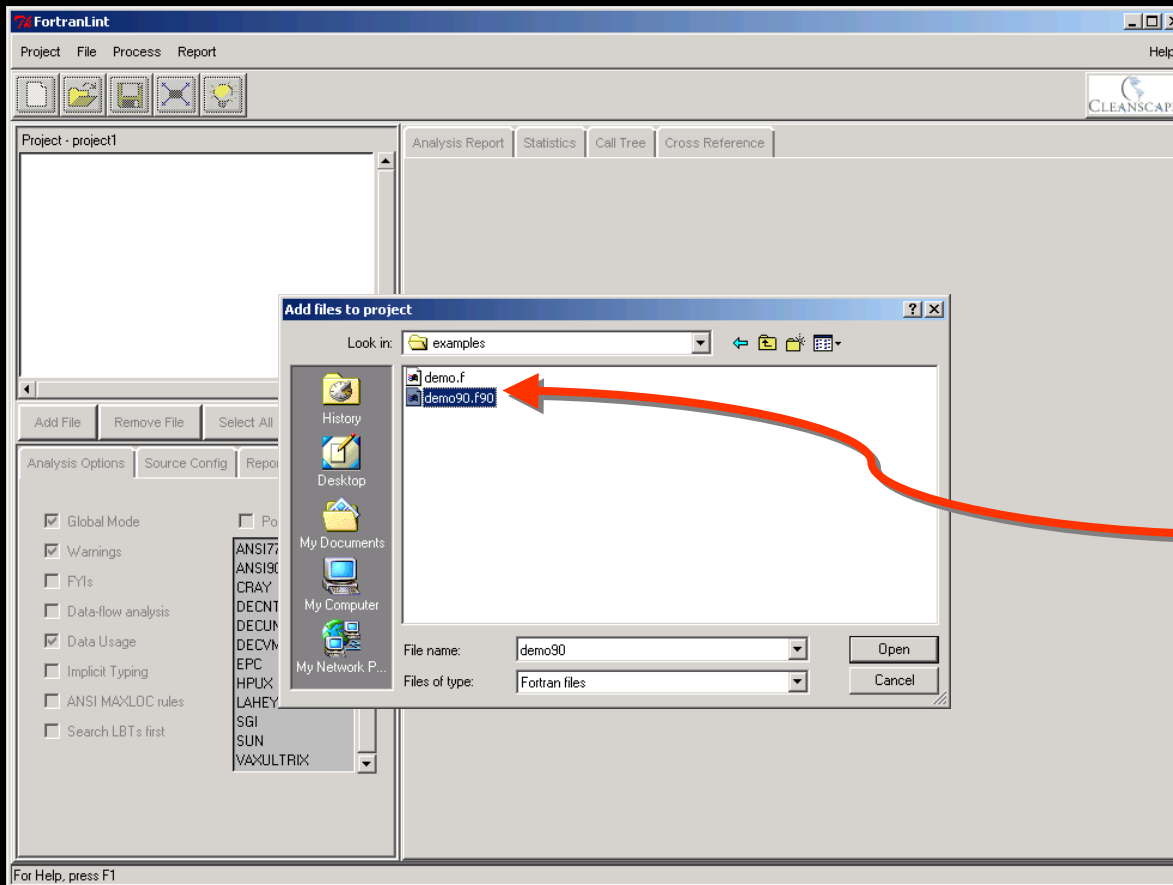
## Add File to Project 1/4



◆ Press “Add File”

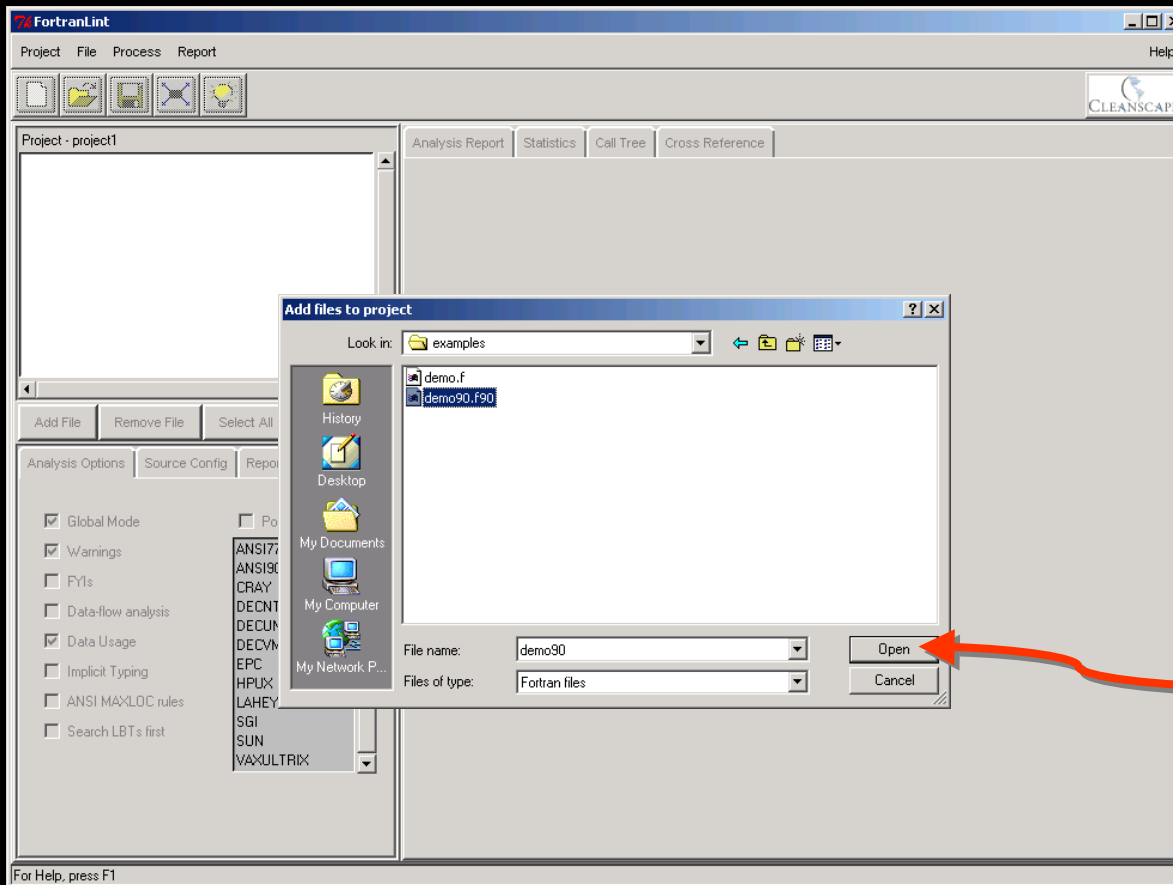
- ◆ Or select:  
File > Add file

## Add File to Project 2/4

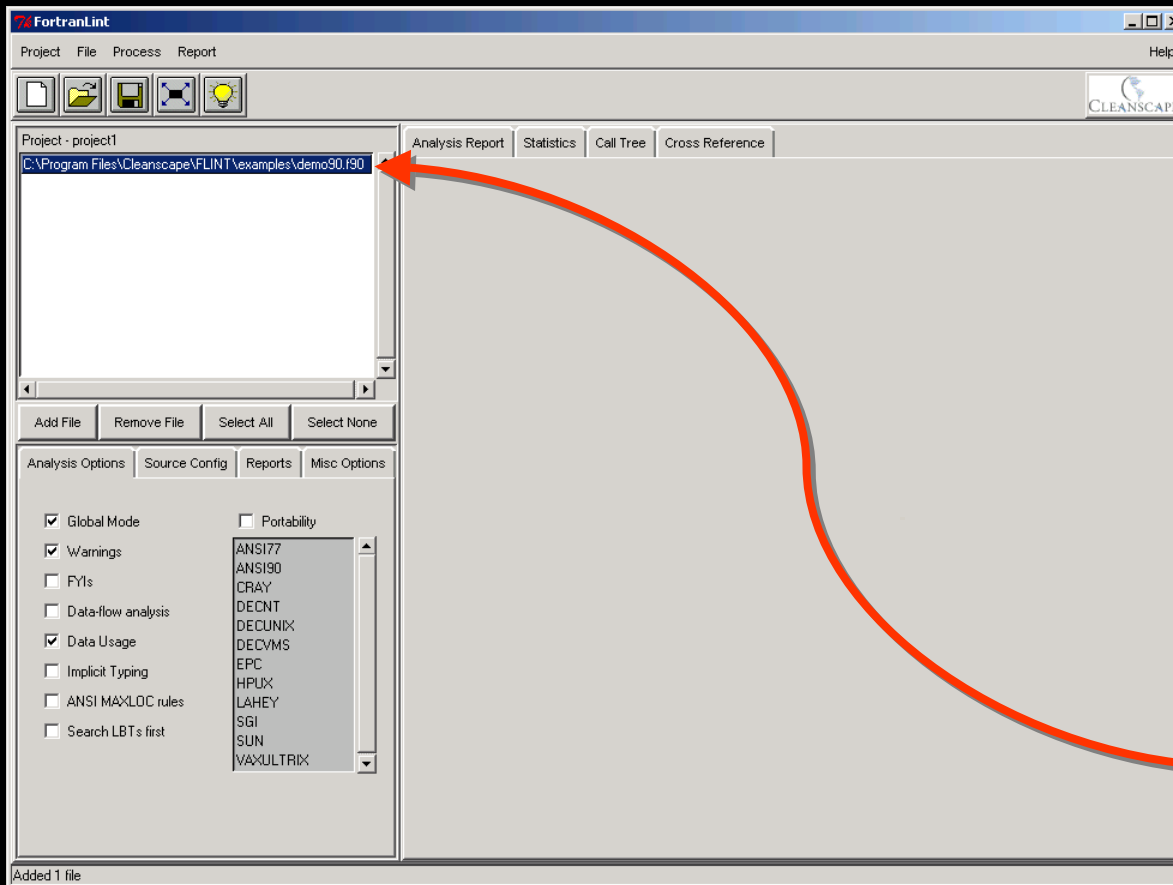


- ◆ Press “Add File”
  - ◆ Or select:  
File > Add file
- ◆ Select file(s) to be analyzed

## Add File to Project 3/4



- ◆ Press “Add File”
  - ◆ Or select:  
File > Add file
- ◆ Select file(s) to be analyzed
- ◆ Click “Open”

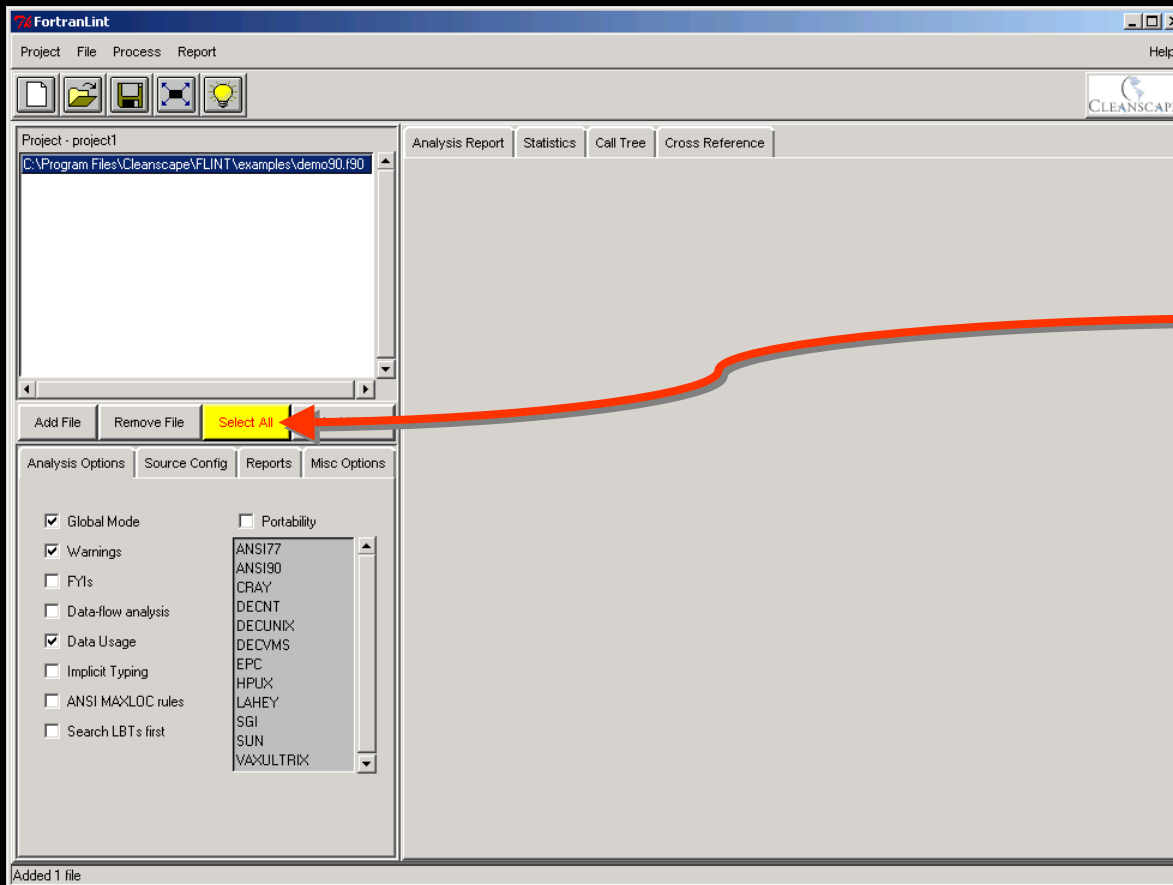


## Add File to Project 4/4

- ◆ Press “Add File”
  - ◆ Or select:  
File > Add file
- ◆ Select file(s) to be analyzed
- ◆ Click “Open”
- ◆ View the project files in Project window



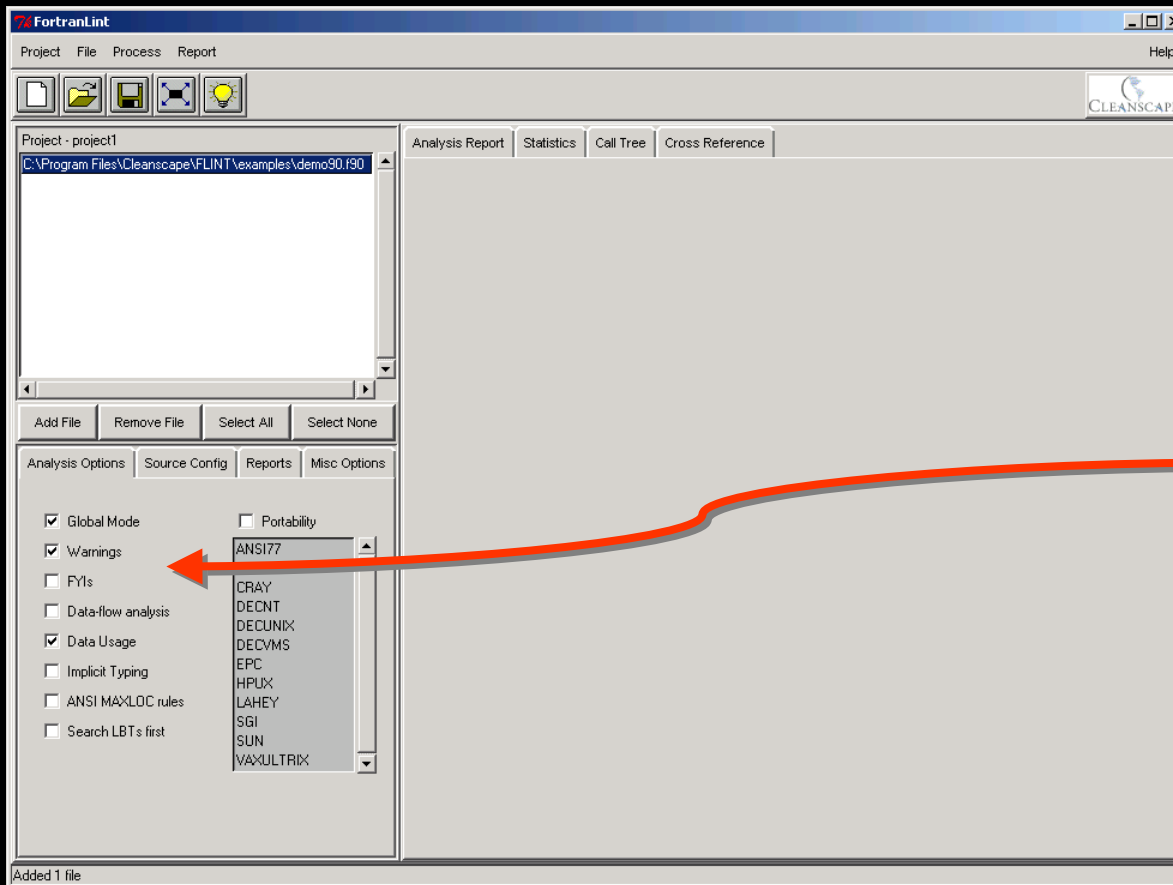
# Analyze your source file(s) 1/5



◆ Select source file(s) to be analyzed

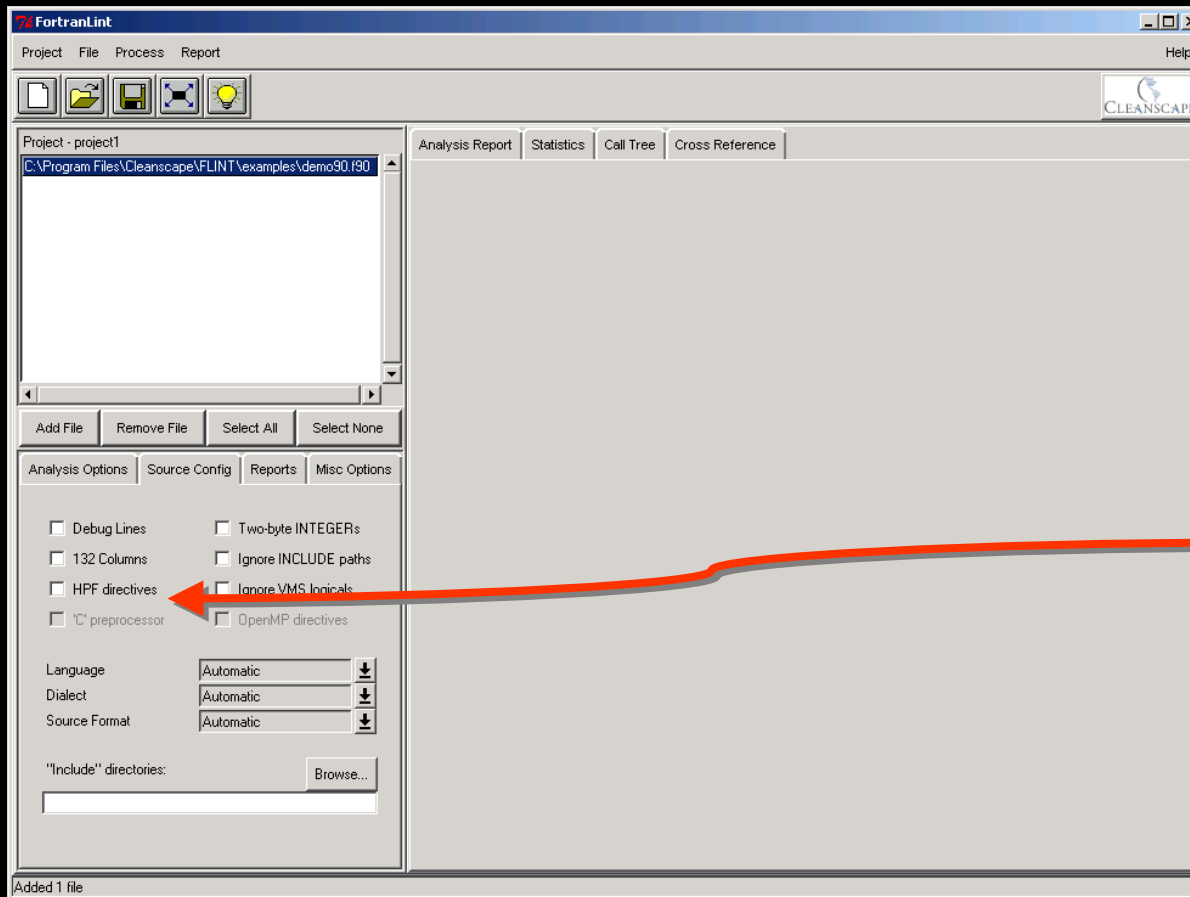
## Analyze your source file(s) 2/5

- ◆ Select source file(s) to be analyzed
- ◆ Select Test Options

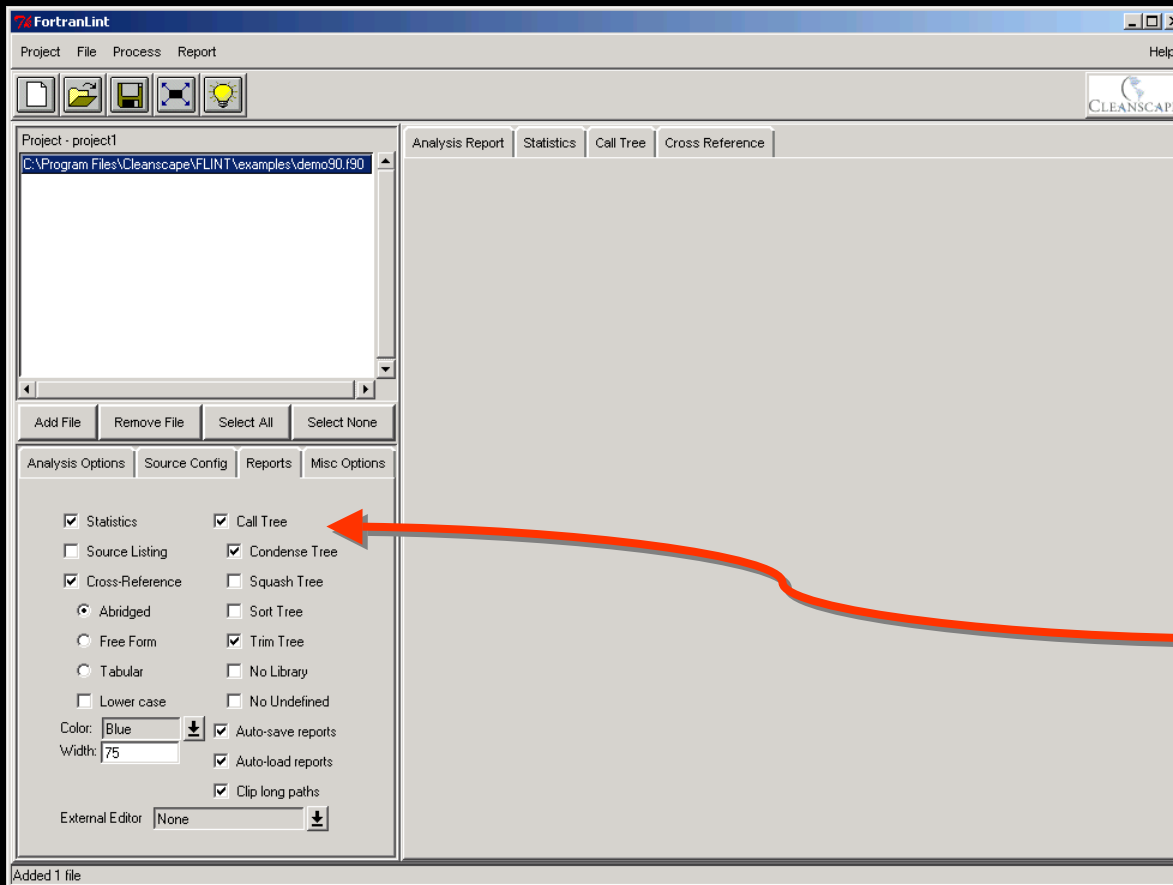


## Analyze your source file(s) 3/5

- ◆ Select source file(s) to be analyzed
- ◆ Select Test Options
- ◆ Select Source Options



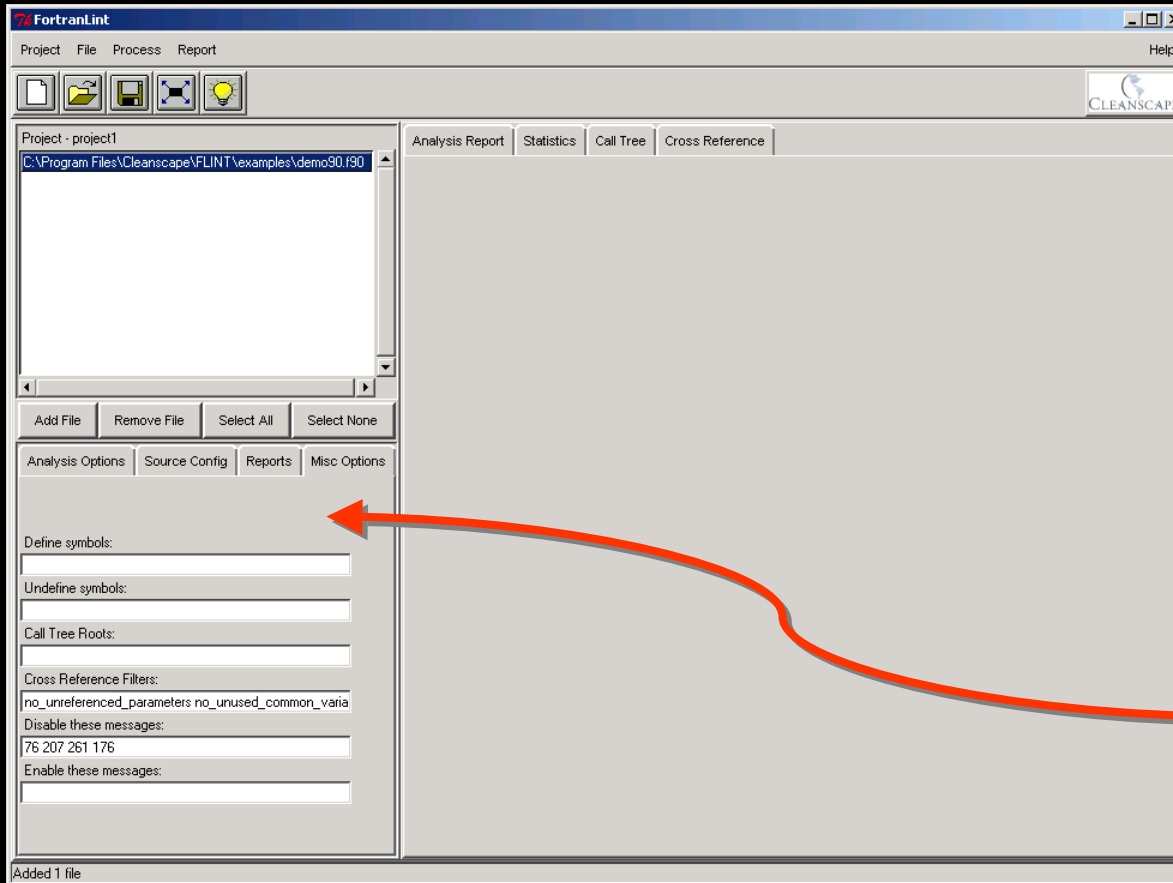
## Analyze your source file(s) 4/5

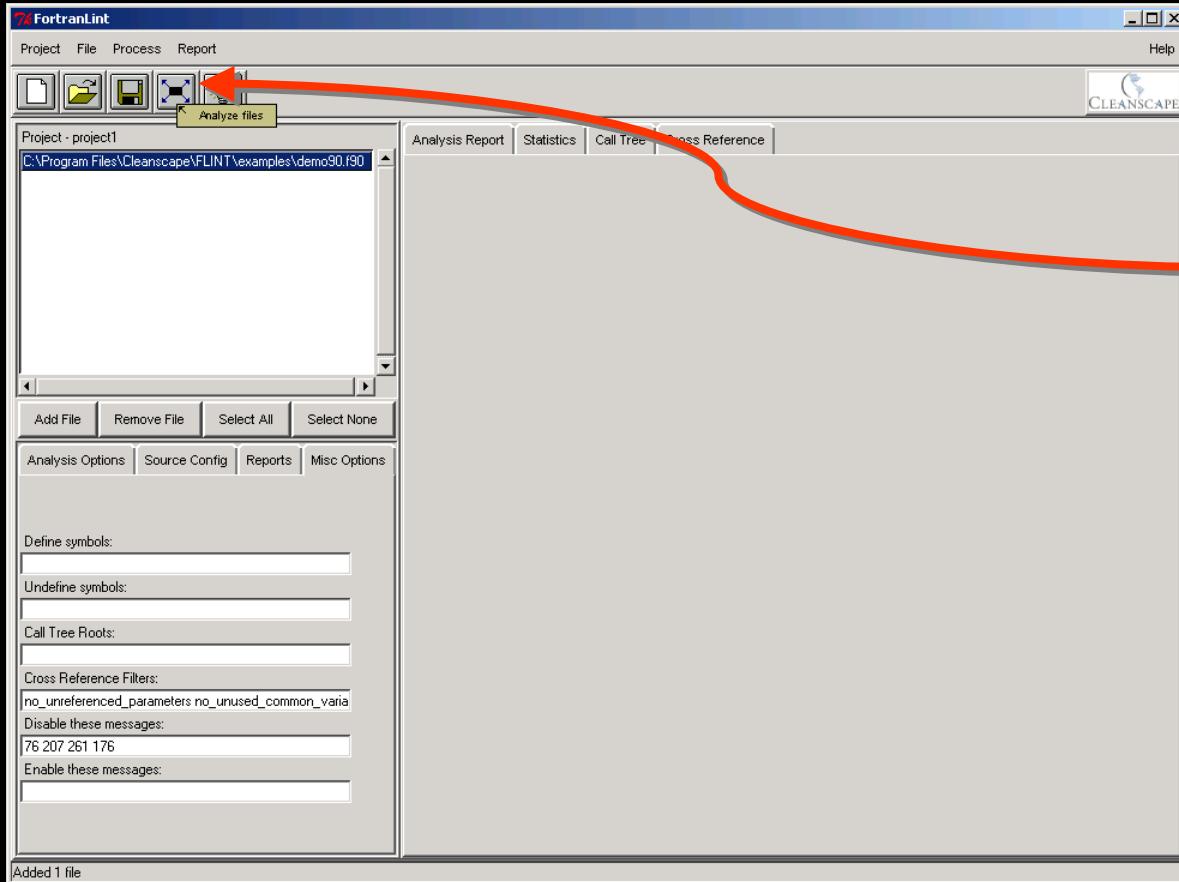


- ◆ Select source file(s) to be analyzed
- ◆ Select Test Options
- ◆ Select Source Options
- ◆ Select Report Options

## Analyze your source file(s) 5/5

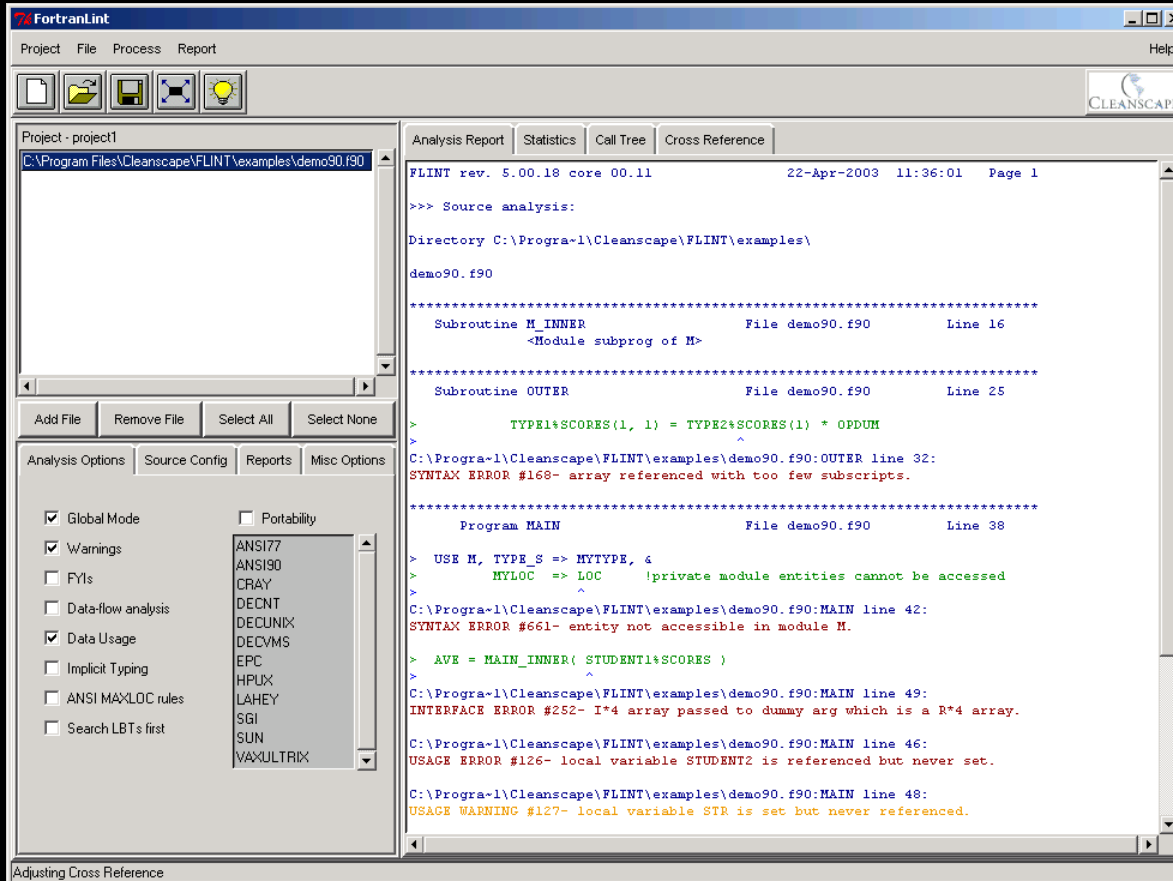
- ◆ Select source file(s) to be analyzed
- ◆ Select Test Options
- ◆ Select Source Options
- ◆ Select Report Options
- ◆ Select Miscellaneous Options





## Execute test 1/2

◆ Push Execute Test button



## Execute test 2/2

- ◆ Push Execute Test button
- ◆ Don't blink!
  - ◆ Almost instantaneous processing for 100,000 lines of code
  - ◆ A few seconds for 1 million lines of code

## View Reports 1/4

The screenshot shows the FortranLint application window. The main window displays the following analysis report:

```

FLINT rev. 5.00.18 core 00.11          22-Apr-2003  11:36:01  Page 1

>>> Source analysis:

Directory C:\Progra-1\Cleanscape\FLINT\examples\
demo90.f90

-----
Subroutine M_INNER          File demo90.f90      Line 16
  <Module subprog of M>
-----
Subroutine OUTER          File demo90.f90      Line 25
>
  TYPE1+SCORES(1, 1) = TYPE2+SCORES(1) * OPDUM
  ^
C:\Progra-1\Cleanscape\FLINT\examples\demo90.f90:OUTER line 32:
SYNTAX ERROR #168- array referenced with too few subscripts.

-----
Program MAIN          File demo90.f90      Line 38
>
> USE M, TYPE_S => MYTYPE, &
> MYLOC => LOC !private module entities cannot be accessed
  ^
C:\Progra-1\Cleanscape\FLINT\examples\demo90.f90:MAIN line 42:
SYNTAX ERROR #661- entity not accessible in module M.
>
> AVE = MAIN_INNER( STUDENT1+SCORES )
  ^
C:\Progra-1\Cleanscape\FLINT\examples\demo90.f90:MAIN line 49:
INTERFACE ERROR #252- I*4 array passed to dummy arg which is a R*4 array.
C:\Progra-1\Cleanscape\FLINT\examples\demo90.f90:MAIN line 46:
USAGE ERROR #126- local variable STUDENT2 is referenced but never set.
C:\Progra-1\Cleanscape\FLINT\examples\demo90.f90:MAIN line 48:
USAGE WARNING #127- local variable STR is set but never referenced.

```

The left sidebar shows the 'Analysis Options' tab with the following settings:

- Global Mode
- Warnings
- FYIs
- Data-flow analysis
- Data Usage
- Implicit Typing
- ANSI MAXLOC rules
- Search LBTs first
- Portability
- ANSI77
- ANSI90
- CRAY
- DECNT
- DECUNIX
- DECVMS
- EPC
- HPUX
- LAHEY
- SGI
- SUN
- VAXULTRIX

- ◆ Analysis Report
- ◆ Statistics Report
- ◆ Call Tree Report
- ◆ Cross Reference Report



## View Reports 2/4

FortranLint

Project - project1

C:\Program Files\Cleanscape\FLINT\examples\demo90.f90

Analysis Report | Statistics | Call Tree | Cross Reference

FLINT rev. 5.00.18 core 00.11 22-Apr-2003 11:36:01 Page 1

>>> Statistics:

Number of source files: 1

Source files: 54 lines, 1273 bytes ( 18% comments, 82% code )  
 Include files: 14 lines, 352 bytes ( 5% comments, 95% code )  
 Total parsed: 68 lines, 1625 bytes ( 15% comments, 85% code )

Total subprograms: 5  
 Subroutines: 2  
 Functions: 1  
 Program: 1  
 Block Data: 0  
 Module: 1

Individual message summary

USAGE ERR #126- lx: local variable \* is referenced but never set.  
 USAGE WARN #127- lx: local variable \* is set but never referenced.  
 SYNTAX ERR #168- lx: array referenced with too few subscripts.  
 INTFC ERR #252- lx: \* array passed to dummy arg which is a \* array.  
 SYNTAX ERR #661- lx: entity not accessible in module \*.  
 USAGE WARN #743- lx: module entity set but not referenced: \*, \*

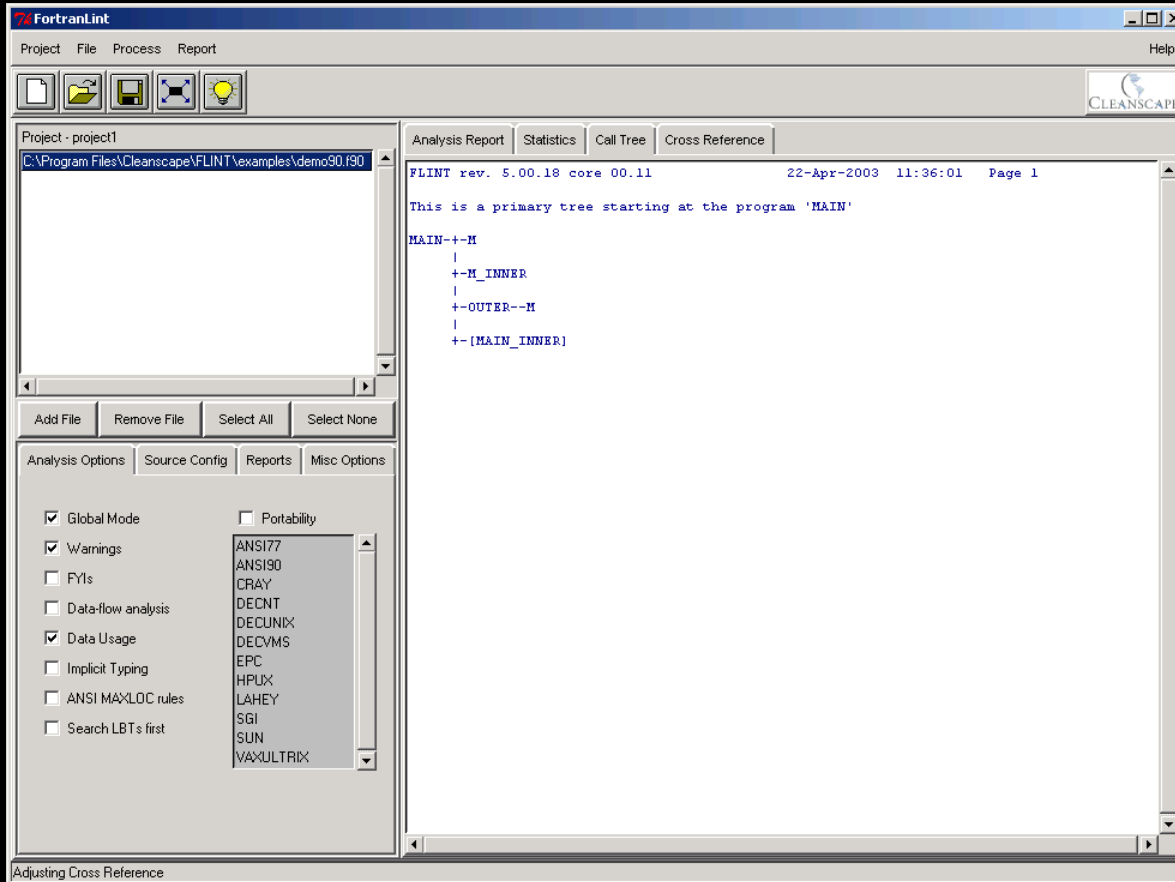
Total messages: 6

	Errors	Warnings	FYIs
Syntax:	2	0	<supp>
Interface:	1	0	<supp>
Data usage:	1	2	<supp>
Implicit typing:			<supp>

Default options: -0276  
 Remanded options: --? -- -n -TC:\Program Files\Cleanscape\FLINT\examples -- --d

Adjusting Cross Reference

- ◆ Analysis Report
- ◆ Statistics Report
- ◆ Call Tree Report
- ◆ Cross Reference Report



## View Reports 3/4

- ◆ Analysis Report
- ◆ Statistics Report
- ◆ Call Tree Report
- ◆ Cross Reference Report

## View Reports 4/4

FortranLint

Project File Process Report Help

Project - project1

C:\Program Files\Cleانسcape\FLINT\examples\demo90.f90

Analysis Report Statistics Call Tree Cross Reference

FLINT rev. 5.00.18 core 00.11 22-Apr-2003 11:36:01 Page 1

\*\*\*\*\* SYMBOL TABLE \*\*\*\*\*

\*\*\* Program:

MAIN : defined at line 38  
 of C:\Program Files\Cleانسcape\FLINT\examples\demo90.f90  
 Calls- demo90.f90:M, demo90.f90:M::M\_INNER,  
 demo90.f90:OUTER,  
 demo90.f90:MAIN::MAIN\_INNER

\*\*\* Subroutines:

M\_INNER : M internal : defined at line 16  
 of C:\Program Files\Cleانسcape\FLINT\examples\demo90.f90  
 Args- (type MYTYPE S, type MYTYPE R)  
 Called by- demo90.f90:MAIN

OUTER : defined at line 25  
 of C:\Program Files\Cleانسcape\FLINT\examples\demo90.f90  
 Args- (R\*4 array R, type MYTYPE R, I\*4 R0)  
 Calls- demo90.f90:M  
 Called by- demo90.f90:MAIN

\*\*\* Functions:

INT : I\*4 : intrinsic function  
 Called by- demo90.f90:MAIN::MAIN\_INNER

KIND : I\*4 : intrinsic function  
 Called by- demo90.f90:MAIN::MAIN\_INNER

MAIN\_INNER : I\*4 : MAIN internal : defined at line 53  
 of C:\Program Files\Cleانسcape\FLINT\examples\demo90.f90  
 Args- (R\*4 array R)  
 Called by- demo90.f90:MAIN

PRESENT : L\*4 : intrinsic function  
 Called by- demo90.f90:OUTER

SIZE : I\*4 : intrinsic function  
 Called by- demo90.f90:MAIN::MAIN\_INNER

FORTRAN-1int (symbol table - Modules) Page 2

Adjusting Cross Reference

Analysis Options Source Config Reports Misc Options

Global Mode  Portability

Warnings

FYIs

Data-flow analysis

Data Usage

Implicit Typing

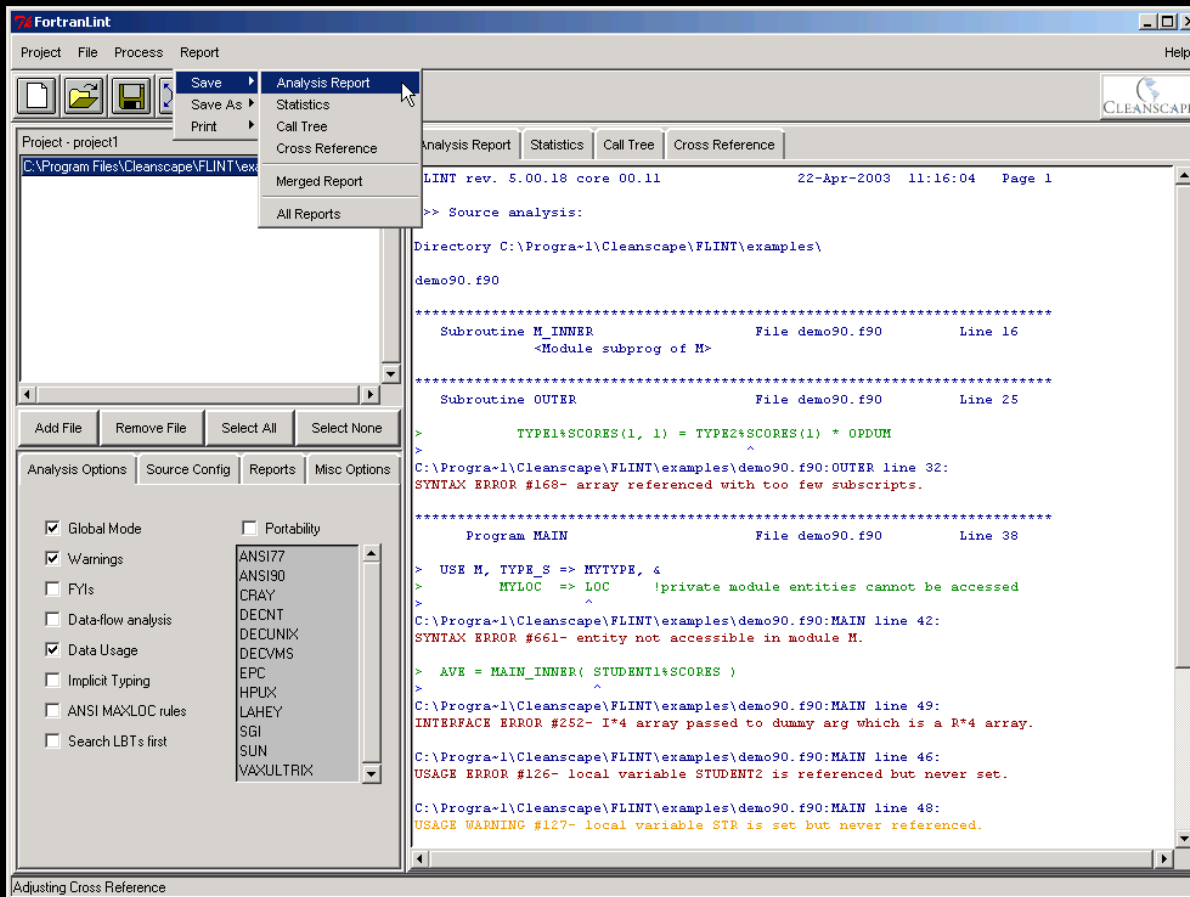
ANSI MAXLOC rules

Search LBT's first

ANSI77  
 ANSI90  
 CRAY  
 DECNT  
 DECLINK  
 DECVMS  
 EPC  
 HPLUX  
 LAHEY  
 SGI  
 SUN  
 VAXULTRIX

- ◆ Analysis Report
- ◆ Statistics Report
- ◆ Call Tree Report
- ◆ Cross Reference Report

# Save & Print Reports



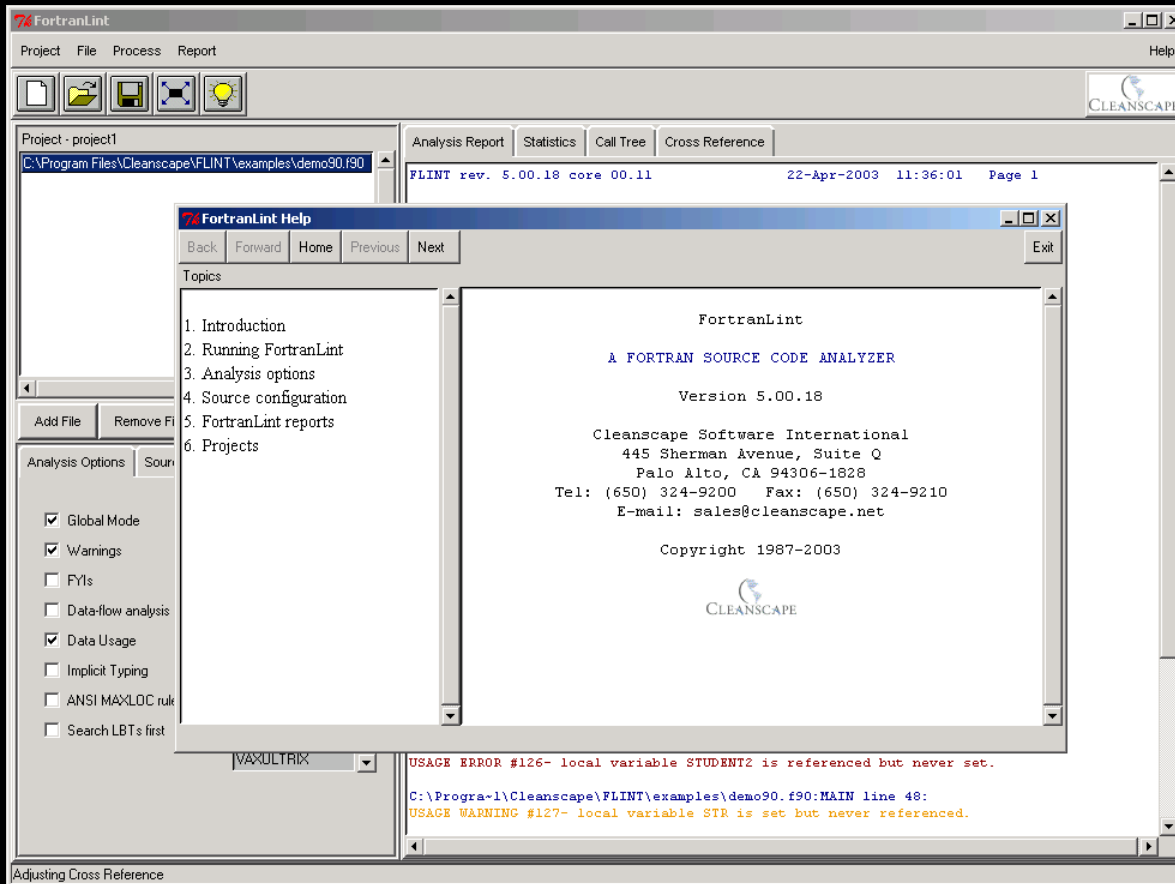
## ◆ Save

- ◆ Select Report > Save
  - Save each report separately
  - Save merged report

## ◆ Print

- ◆ Select Report > Print
  - Print each report separately
  - Print all reports

## Get help



### ◆ Online Help

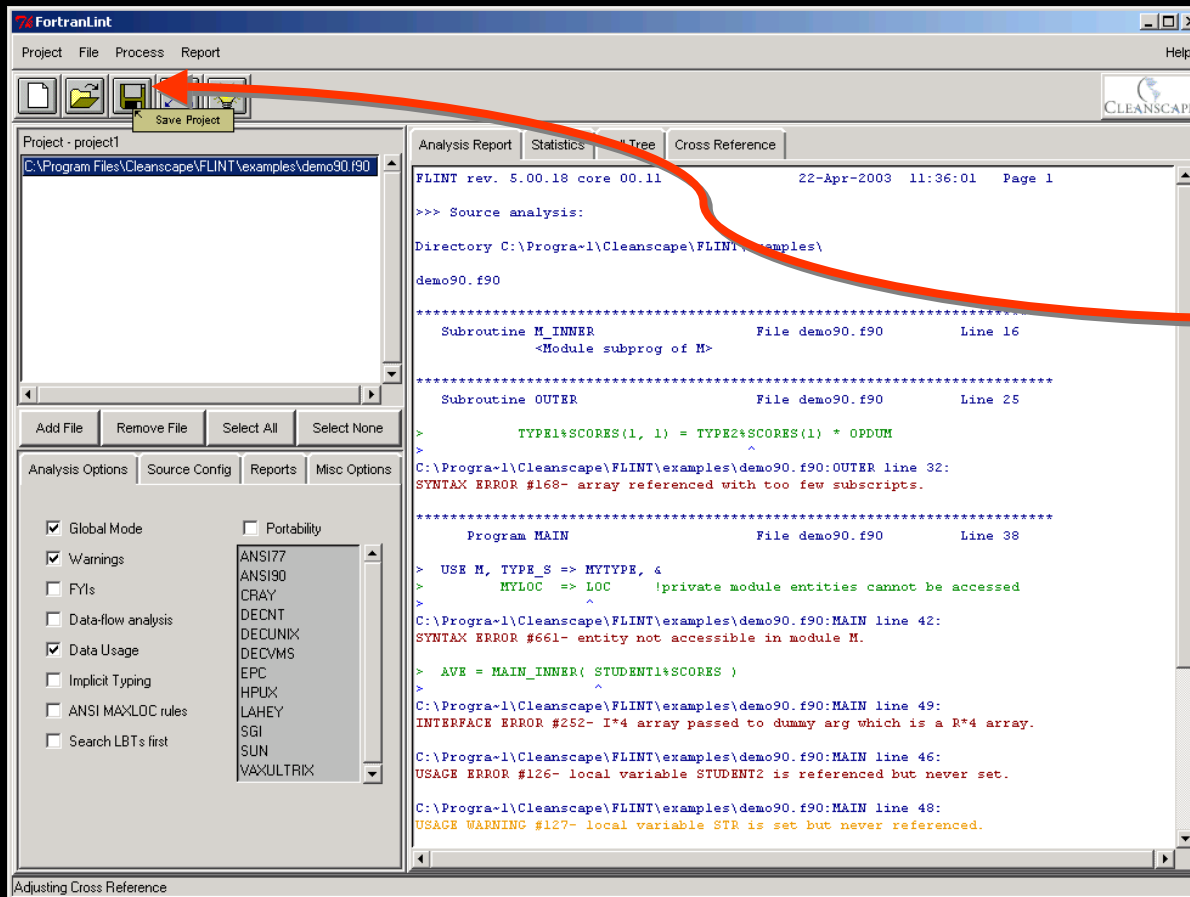
- ◆ Push Help button

### ◆ Documentation

- ◆ Quick Start Guide
- ◆ Users Guide
- ◆ Tutorial

### ◆ Support

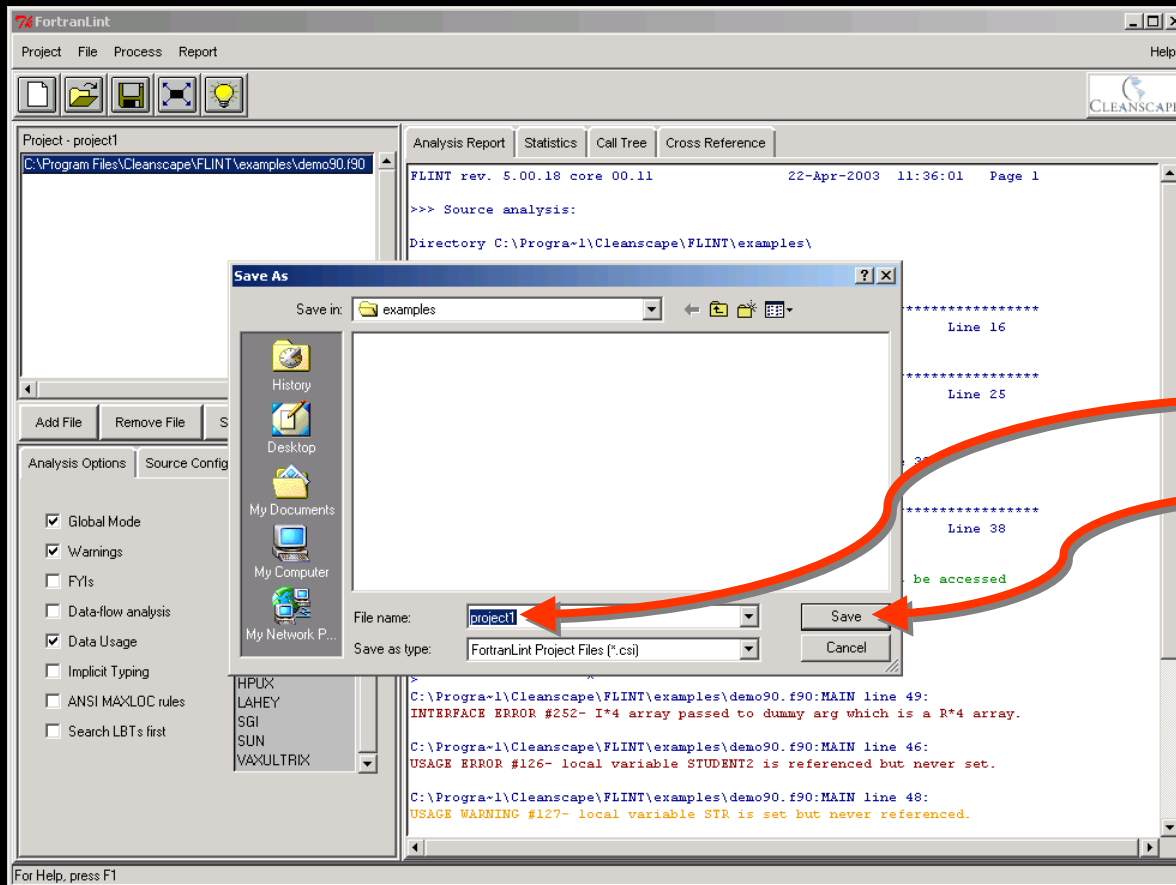
- ◆ <http://www.cleanscape.net>



## Save your project 1/2

- ◆ Push Save button
- ◆ Type project name
- ◆ Push Save

## Save your project 2/2



◆ Push Save button

◆ Type project name

◆ Push Save



*Software Development Powerfully Simplified*

- **Code:** Automatically generate code from patterns
- **Analyze:** Stop problems at the source
- **Test:** Test and visualize software
- **Manage:** Automate the software development process

<http://www.cleanscape.net>

## FortranLint

- ♦ **Detect Fortran bugs and other problems that a compiler can't catch**
- ♦ **Conduct thorough assessment with global call interface checking, local dataflow analysis, and best practices information**
- ♦ **Conduct portability checks between multiple host environments an the ANSI F77/F90 standards**
- ♦ **Rapidly understand Fortran sources via call trees and cross references**