Visual Analyzer V2.1
User’s Guide
Preface

Purpose of This Manual

This manual explains how to use the Visual Analyzer. The Visual Analyzer operates under the following operating systems:

- Microsoft® Windows NT® Operating System, Version 3.51 or 4.0.
- Microsoft Windows® 95 and Windows® 98 Operating System.

Readers of This Manual

This manual is intended for Fortran or C programmers; it explains all the functions of the Visual Analyzer. The Visual Analyzer analyzes Fortran and C code within the range set by ANSI standards and Fujitsu extensions to those standards. You should have a basic knowledge of Windows NT® Version 3.51 or 4.0, Windows® 95 or Windows® 98.

Organization of This Manual

- Chapter 1 -- Outline
  Chapter 1 explains the structure of the Visual Analyzer.
- Chapter 2 -- How to analyze the source program
  Chapter 2 explains how to analyze the source program.
- Chapter 3 -- Call Graph
  Chapter 3 explains the display of the call relationships among program units or functions by Graph.
- Chapter 4 -- Call Tree
  Chapter 4 explains the display of the call relationships among program units or functions by Tree.
- Chapter 5 -- Logic Flow
  Chapter 5 explains the display of the internal structure of a program unit or function.
- Chapter 6 -- Statistical Listing
  Chapter 6 explains the display of the statistical listing about any program unit or function.
- Chapter 7 -- Argument Listing
  Chapter 7 explains the display and checking argument listing passed between any of the program units or functions.
- Chapter 8 -- Cross-Reference Listing
  Chapter 8 explains the display of the Cross-Reference Listing about any program unit or function.
Chapter 9 -- Global Listing
Chapter 9 explains the display of the C global variables and Fortran common blocks listing.

Chapter 10 -- Inspection of a program
Chapter 10 explains how to determine if the code is non-standard conforming, or contains a conflict argument between actual and dummy arguments.

Chapter 11 -- Printing
Chapter 11 explains how to print analyzed results.

Notes on this Manual
The examples in this manual are based on Windows NT® Version 4.0. This manual contains programming service information, which can be used to analyze application programs that use Fortran or C language.
# Table of Contents

**Preface** ................................................................................................................................. 2
Purpose of This Manual .................................................................................................................. 2
Readers of This Manual .................................................................................................................. 2
Organization of This Manual ......................................................................................................... 2
Notes on this Manual ....................................................................................................................... 3

**Chapter 1 -- The Visual Analyzer** ........................................................................................ 10
Outline of Functions ..................................................................................................................... 10
  - Call Graph ................................................................................................................................ 10
  - Call Tree ..................................................................................................................................... 11
  - Logic Flow .................................................................................................................................. 11
  - Statistical Listing ....................................................................................................................... 11
  - Argument Listing ....................................................................................................................... 12
    - Fortran Program Units ............................................................................................................. 12
    - C Functions ............................................................................................................................ 12
  - Cross-Reference Listing ............................................................................................................ 12
    - Fortran Program Units ............................................................................................................. 12
    - C Functions ............................................................................................................................ 13
  - Global Listing ........................................................................................................................... 13
    - Fortran Program Units ............................................................................................................. 13
    - C Functions ............................................................................................................................ 13
  - Inspection of a Program ............................................................................................................ 13
  - Printing ..................................................................................................................................... 13

**Chapter 2 -- How to Analyze the Source Program** .................................................................. 14
Activating the Visual Analyzer ..................................................................................................... 15
Terminating the Visual Analyzer .................................................................................................. 15
The Main Window of the Visual Analyzer .................................................................................... 16
The Menu of the Main Window of the Visual Analyzer .................................................................. 17
  - Toolbar ...................................................................................................................................... 18
  - Status Bar ................................................................................................................................. 18
  - The File Menu ......................................................................................................................... 18
    - New Project Command ......................................................................................................... 18
    - Open Project Command ........................................................................................................ 18
    - Save Project Command .......................................................................................................... 19
    - Close Project Command ........................................................................................................ 19
    - Delete Project Command ...................................................................................................... 19
    - Print Command ..................................................................................................................... 19
    - Print Preview Command ...................................................................................................... 19
    - Printer Setup Command ....................................................................................................... 19
    - Exit Visual Analyzer Command ......................................................................................... 19
  - Analyze Menu ....................................................................................................................... 20
Start Analysis Command ................................................................. 20
Stop Analysis Command ............................................................... 20
View Call Graph Command .......................................................... 20
Function Name List Command ...................................................... 20
View Call Tree Command ............................................................ 20
Select Files Command ................................................................. 20
Options Command ........................................................................ 20
Edit Menu ...................................................................................... 21
Copy Command ........................................................................... 21
Select All Command ..................................................................... 21
Find Command ............................................................................ 21
Find Previous Command ............................................................. 22
Find Next Command .................................................................... 22
View Menu .................................................................................. 22
Toolbar Command ........................................................................ 22
Status Bar Command ................................................................. 22
Check Menu ................................................................................ 23
Argument Command .................................................................... 23
Common Block Command .......................................................... 23
Fortran Standard Command ....................................................... 23
Global Menu ................................................................................. 23
Global (C) Command .................................................................. 23
Common Definition Command .................................................. 24
Common Reference Command .................................................. 24
Module Definition Command ..................................................... 24
Module Reference Command ...................................................... 24
Window Menu .............................................................................. 24
Cascade Command ....................................................................... 24
Tile Horizontally Command ....................................................... 24
Tile Vertically Command ........................................................... 25
Arrange Icons Command ............................................................ 25
Close All Command ..................................................................... 25
Help Menu ................................................................................... 25
Help Topics Command ............................................................... 25
About Visual Analyzer Command ............................................... 25
Selecting the Analysis Source File ................................................ 25
Specifying Analysis Options ....................................................... 27
Specifying Common Options for Fortran and C Programs .......... 29
  Output Directory ..................................................................... 29
  Browse Button ..................................................................... 29
Specifying Fortran Options ........................................................ 29
  Format ................................................................................... 29
  Free ..................................................................................... 29
  Fixed ..................................................................................... 29
Close All Command................................................................. 51
Recursive Command ................................................................. 51
Detail Command ........................................................................ 51
All Functions Command ............................................................ 51
Find Command ........................................................................... 51
Find Caller Command .................................................................. 51
Linkage Command ..................................................................... 51
Displaying Undefined Program Units or Functions....................... 52
Find Function............................................................................... 52

Chapter 5 -- Logic Flow................................................................ 53
Call Graph Display ...................................................................... 53
Logic Flow Popup Menu .............................................................. 55
Comment Command .................................................................... 55
User Include Command ............................................................... 55
System Include Command .......................................................... 55
Find Command ........................................................................... 55
Save As Source File Command ................................................... 55
Logic Flow Display ..................................................................... 56
Fortran Program Logic Flow Display ......................................... 56
C Program Logic Flow Display .................................................. 59
Searching for Names .................................................................. 63
Fortran Search Targets ............................................................... 63
C Search Targets ....................................................................... 63

Chapter 6 -- Statistical Listing .................................................. 64
Popup Menu of Statistical Listing ................................................. 65
Copy Command .......................................................................... 66
Select All Command ................................................................... 66
Logic Flow Command .................................................................. 66
Find Command ........................................................................... 66
Save AS Text File Command ....................................................... 66
Displaying Statistical Information ................................................ 66
Displaying Statistical Information For Fortran Programs .............. 66
Displaying Statistical Information For C Programs ..................... 69
Measuring the Complexity of Fortran Programs ......................... 72
Measuring the Complexity of C Programs ................................... 72
Interpreting Complexity Values ................................................... 73
Searching for Character Strings .................................................. 73

Chapter 7 -- Argument Listing .................................................. 74
Popup Menu of Linkage Listing .................................................... 75
Copy Command .......................................................................... 76
Select All Command ................................................................... 76
Logic Flow Command .................................................................. 76
Find Command ........................................................................... 76
Save As Text File Command ....................................................... 76
Chapter 1 -- The Visual Analyzer

The Visual Analyzer performs static analysis of a program written in Fortran or C and provides visual displays of the analysis results. These displays allow you to check the program structure and the status of data within the program.

This chapter explains the functions and procedures of the Visual Analyzer.

The Visual Analyzer can fulfill the following user requirements:

- To create on the fly diagrams and other immediate visual aids so that you can see the broad and detailed flow of your program without reading an annotated listing.
- To identify where code in other programs needs to be changed as a result of changes to common variables (Fortran) or global variables (C).
- To document the program.

Outline of Functions

This chapter outlines each Visual Analyzer function.

The Visual Analyzer performs the following nine functions:

- Call Graph (Displays the call relationships among program units and functions by Graph)
- Call Tree (Displays the call relationships among program units and functions by Tree)
- Logic Flow (Displays the structure of individual program units or functions)
- Statistical Listing (Displays statistical information for individual program units or functions)
- Argument Listing (Displays the actual arguments and dummy arguments passed between program units or functions)
- Cross-Reference Listing (Displays the cross-reference information for individual program units or functions)
- Global Listing (Displays the global data information, global variables in C, common blocks, or Module subprograms in Fortran)
- Determine if a program is a Fortran standard-conforming program.
- Printing (Prints the above Listing)

Call Graph

Call Graph displays the call relationships among program units or functions by Graph. Call Graph saves the image as the bitmap data. Call Graph has the following five functions:

- Displays the entire structure of an analyzed program
- Displays the call relationships from a specific program unit or function
Displays the call relationships around a specific program unit or function
Displays the program units or functions which are called from some functions, but which are not defined in the analyzed source programs
Displays the number of lines of each program unit or function

Call Tree

Call Tree displays the call relationships among program units or functions by Tree. Call Tree saves as text data. Call Tree has the following five functions:
Displays the entire structure of an analyzed program
Displays the call relationships from a specific program unit or function
Displays the call relationships around a specific program unit or function
Displays the program units or functions which are called from some functions, but which are not defined in the analyzed source programs
Searches program unit or function name

Logic Flow

Logic Flow is the function that displays the structure of individual program units or functions. Logic Flow of Fortran saves as the text file. Logic Flow has the following six functions.
For Fortran program code, the Visual Analyzer displays IF, DO, CASE, WHERE, CYCLE, GO TO and EXIT statements using intuitive, color graphical outlines.
For C functions, the Visual Analyzer displays if, for, switch, break, goto, while, continue and return statements, and structure and union specifiers using intuitive, color graphical outlines.
Controls the display of comment lines
Controls the display of include files
For Fortran program units, searches for variable or procedure names.
For C functions, searches for variable, function, tag, or macro names.

Statistical Listing

Statistical Listing displays the following information for individual program units or functions.
Complexity
Number of statements
Number of included statements
Called program unit or function name
Caller program unit or function name
Appearance count of each statement or label

Argument Listing

The Argument Listing displays the following information for actual arguments and dummy arguments passed between program units or functions

Fortran Program Units

For the actual argument, the information displayed is:
- The source filename
- The line number calling the program unit
- The message indicating the abnormal case resulting from the integrity check (the number of arguments, type) between the actual and dummy argument
- The name, type and order of each actual argument.
- For the dummy argument, the information displayed is the order, name, kind, type, and the line numbers of declaration, setting and reference of each dummy argument.

C Functions

For the actual argument, the information displayed is:
- The source filename
- The line number calling the function
- The message for abnormal case of integrity check (the number of arguments, type) between actual argument and dummy argument, the type of the return value
- The order, name, kind and type of each actual argument.

For the dummy argument, the information displayed is the order, name, kind, type, and the line numbers of declaration, setting and reference of each dummy argument.

Cross-Reference Listing

Cross-Reference Listing displays the following information about the individual program units or functions.

Fortran Program Units

- The line number for the definition, declaration, setting, or reference of a variable or a procedure name.
- The line number of the statement which defines or references the derived type name.
- The line number for the macro definition, the macro name and replacement character, and the invalidated line number of the macro definition.
The line number of the statement which defines the statement label and by which the statement label is referenced.

C Functions

- The line number for the definition, declaration, setting, or reference of a variable or function name.
- The line number for the definition of the tag name of the structure, union or enumerator and typedef name.
- The line number for the macro definition, the macro name and replacement character, and the invalidated line number of the macro definition.

Global Listing

Global Listing displays the following global data information.

Fortran Program Units

- Displays the Fortran common block information.
- Displays the Fortran Module subprogram relationships.

C Functions

- Displays the name and type of the global variables and the function names which use these variables.

Inspection of a Program

You can inspect the following in the analyzed program:

- Inconsistency between actual arguments and dummy arguments.
- The size of Fortran common blocks.
- Inspection of the Fortran 95 standard.
- Verification of the C program.

Printing

You can print graphical charts in hard copy format.
Chapter 2 -- How to Analyze the Source Program

This chapter explains how to analyze the source program.

The figure titled *Relationship of Visual Analyzer Function* shows the relationship of the various functions of the Visual Analyzer.

---

**Relationship of Visual Analyzer Function**

Main Window of Visual Analyzer

- List Window Display
  - See Function Name List
  - Call Graph Display
  - Call Tree Display

Logic Flow Display
- See Logic Flow
- Statistical Information Display
  - See Statistical Information
  - Argument Information Display
    - See Argument Information
  - Cross Reference Information Display
    - See Cross Reference Information
  - Global Information Display
    - See Global Information
  - Inspection of a Program Information Display
    - See Inspection of a Program
Activating the Visual Analyzer

To activate the Visual Analyzer, click the Visual Analyzer icon. The main window of the Visual Analyzer appears. See The Main Window of the Visual Analyzer.

Terminating the Visual Analyzer

To terminate the Visual Analyzer, click the Exit Visual Analyzer command in the File menu.
The Main Window of the Visual Analyzer

After the Visual Analyzer is launched, the main window of the Visual Analyzer appears. The figure titled *Visual Analyzer Main Window* shows the main window of the Visual Analyzer.

*Visual Analyzer Main Window*

After the source program is analyzed, the main window of the Visual Analyzer changes as in the figure titled *The Main Window of the Visual Analyzer After Analysis.*
The Main Window of the Visual Analyzer After Analysis

The Menu of the Main Window of the Visual Analyzer

This section explains the bars and menus in the main window of the Visual Analyzer.
Toolbar

Icons on the Toolbar provide shortcuts for executing commonly used commands. All Toolbar commands can also be executed from the Menu Bar commands.

Status Bar

The name of the currently active or currently selected command appears in the status bar at the bottom of the window.

The File Menu

The file menu contains basic file manipulation and modification commands.

New Project Command

Use this command to display the New Project dialog box and specify the project name. The project directory specified name is created. And project file specified name.pvr is created in the project directory. The project file contains the analysing source file names and analysis options information. The analysis results data file is created in the project directory.

Open Project Command

Use this command to display the Open Project dialog box and specify the existing project file (*.pvr).
Save Project Command
Use this command to save the current opened project to the project file.

Close Project Command
Use this command to close the project the current opened project.

Delete Project Command
Use this command to display the Delete Project dialog box and specify the project file (*.pvr) that user want to delete. When the project file is deleted, the analysis result data files are also removed.

Print Command
Use this command to display the Print dialog box and print the information in the active window. See Printing.

Print Preview Command
Use this command to display a page as it will look when printed.

Printer Setup Command
Use this command to set printer information.

Exit Visual Analyzer Command
Use this command to exit from the System.
Analyze Menu

Start Analysis Command

Use this command to analyze the source file or files specified in the Select File dialog box. See Analyzing a Program.

Stop Analysis Command

Use this command to stop the currently active analysis. The Stop Analysis Command can only be clicked during an analysis of the source file.

View Call Graph Command

Use this command to display the Call Graph view of the source files under analysis.

Function Name List Command

In Call Graph you'll see Function Name List as part of this menu. Use the Function Name List command to list analyzed program units, function names and source filenames. See Function Name List.

View Call Tree Command

Use this command to display the Call Tree view of the source files under analysis.

Select Files Command

Use this command to display the Select File dialog box and specify the source files to be analyzed.

Options Command

Use this command to set the analysis options. See Specifying Analysis Options
Edit Menu

Copy Command
Use this command to copy highlighted text to the Windows clipboard.

Select All Command
Use this command to select (highlight) the entire contents of the active window.

Note
The Edit Menu can be used only during a display of the analysis information. The Copy and Select all commands cannot be used when the Call Graph, Call Tree, Logic Flow or Function name list window is active.

Find Command
Use the Find command to display the Find dialog box and search the text displayed in the active window.

Note
The Function Name List, Call Tree and Find commands can be used only when displaying the analysis information. The Find command cannot be used when the Function Name List window is active.

The Find dialog box looks generally like this (some contexts present a slightly different interface but the functionality is the same):
Visual Analyzer V2.1 User’s Guide—Page 22

Find Previous Command

Use this command to search for the previous instance of the text displayed in the Find What field. The search is limited to the contents of the active window.

Find Next Command

Use this command to search for the next instance of the text displayed in the Find What field. The search is limited to the active window.

View Menu

Toolbar Command

Use the Toolbar command to display the Visual Analyzer Toolbar. A checkmark to the left of the Command name indicates that the Toolbar is visible.

Status Bar Command

Use the Status Bar command to display the Status Bar which provides information about your current command and other environmental topics.
Check Menu

Argument Command

Use this command to perform an argument check (the argument type, the argument kind, the number of arguments, and the function return value) for all the program units and functions. See Checking Inconsistencies Between Actual Arguments and Dummy Arguments.

Common Block Command

Use this command to perform a common size check for all common blocks. See Checking the Size of Fortran Common Blocks.

Fortran Standard Command

Use this command to check Fortran programs to determine if they are compliant with the Fortran 95 standard. See Inspection of Fortran Conformance Testing (all).

Global Menu

Global (C) Command

Use this command to list the global variables in C programs and the function names which use those variables. See Displaying C Global Variables.
Common Definition Command

Use this command to display, for each common block, the common block size, the common block element names, the element types, and the program names using that common block. See Displaying Fortran Common Blocks.

Common Reference Command

Use this command to display, for each program, the program name and the common block names which are used in that program. See Displaying Fortran Common Blocks.

Module Definition Command

Use this command to display, for each module subprogram, the module subprogram names and the program names which use that module. See Displaying Fortran Module Subprogram Relationships.

Module Reference Command

Use this command to display, for each program, the program name and the module subprogram names which are used in that program. See Displaying Fortran Module Subprogram Relationships.

Window Menu

Cascade Command

Use this command to display windows in a cascaded fashion so that all open Visual Analyzer windows overlap revealing the Title Bar for each window.

Tile Horizontally Command

Use this command to arrange windows in a horizontal fashion so that all open unminimized Visual Analyzer windows are resized to fit next to each other.
Tile Vertically Command
Use this command to arrange windows in a vertical fashion so that all open Visual Analyzer windows can be sized to fit on top of one another.

Arrange Icons Command
Use this command to arrange all the icons along the bottom of the Visual Analyzer window.

Close All Command
Use this command to close all the Visual Analyzer windows at once.

Help Menu
The Help Menu displays online help information about the Visual Analyzer.

Help Topics Command
Use this command to access the Visual Analyzer online help system.

About Visual Analyzer Command
Use this command to display the current version level of the Visual Analyzer.

Selecting the Analysis Source File
This section explains how to select the source file(s) to be analyzed using the Select Files command. When the Select Files command in the Analyze menu is clicked, the Select Files dialog box as shown in the figure titled Select Files Dialog Box and Selected File List Window are displayed.
1. Select the directory which contains source files in the Directories and the Drives boxes.
2. Click the analysis source filename in the Select File Dialog box and click the OK button, or double-
click the analysis source filename in the Select File Dialog box. When you want to select the multiple analysis source filenames, click them pushing Shift or Ctrl key.

3. The selected source filename is shown in the Selected File List Window.

4. To delete the selected source filenames, click the source filename in the Selected File List Window and click the mouse right button.

Specifying Analysis Options

Several options can be specified for the analysis of source files. This section explains how to specify the analysis options using the Options dialog box.

Invoke the Options command from the Analyze menu.
Options Dialog Box
Specifying Common Options for Fortran and C Programs

Output Directory

To specify the directory in which the Visual Analyzer generates the analysis results data file, click on the [Default directory] or [Change directory] radio button. When the [Default directory] radio button is clicked, it produces the analysis results data file in the default directory listed beside the radio button. When the [Change Directory] radio button is clicked, the directory name field under the radio button is allowed to enter. When the [Browse] button is clicked, the directory where the Visual Analyzer is going to generate the analysis results data file can be changed. See Analysis results data file.

The default output directory is the project directory.

Browse Button

Select the directory into which the Visual Analyzer puts the analysis results data file.

Specifying Fortran Options

![The Fortran Program Options Dialog Box]

The following Fortran program options can be specified.

Format

This option indicates the source format of the Fortran program.

Free

The program is interpreted as a free format Fortran source program.

Fixed

The program is interpreted as a fixed format Fortran source program.
Default

Depending on the suffix of the analyzed Fortran source filename, its default source format is selected automatically as follows:

- If the suffix is .f or .for, its source program is interpreted as fixed format.
- If the suffix is .f90 or .f95, its source program is interpreted as free format.

Other Fortran Options

The following Fortran options can be specified in the Other Options input field:

- **-I path**
  
  Include Path
  
  Default: current directory
  
  Specify -I path to instruct the Visual Analyzer to search the specified path(s) for Fortran INCLUDE files after searching the current directory. Separate multiple search paths with a semicolon, no spaces.

  **Example**
  
  `-i ..\project2\includes;..\project3\includes`

  In this example, the Visual Analyzer first searches the current directory, then searches ..\projects\includes and finally ..\project3\includes for INCLUDE files specified in the analysis source file(s).

- **-[N]MAXFATALS number**
  
  Maximum Number of Fatal Errors
  
  Specify -maxfatales to limit the number of fatal errors Visual Analyzer will generate before aborting.

- **-[N]SWM msg**
  
  Suppress Warning Message(s)
  
  Default: -nswm
  
  To suppress a particular error message, specify its number after -swm.

  **Example**
  
  `-swm 16,32`
  
  This example would suppress warning messages 16 and 32. To suppress all warnings, use -nw.

- **-[N]W**
  
  Warn
  
  Default: -w
  
  Specify -w to generate messages that warn of possible programming errors that are not serious enough to prevent execution.
Specifying C Options

The following C program options can be specified.

Lang Level

This option indicates the level of language specification.

K&R

The C language specifications used at analysis time include C language standard (ANSI standard) specifications that are compatible with the conventional (pre-ANSI standard) specifications. If it encounters a program structure in which the operation is different between the old and new language specifications, the Visual Analyzer (including the preprocessor) generates a warning message and interprets it according to the old specifications. For example, a warning message is generated for three-letter expressions (trigraph sequences) and \a and other escape sequences, as well as for items related to changes in the rules for integer expansion.

Extended ANSI

The C language specifications used at analysis time include all of the new specifications of the C language standard (ANSI standard). The interpretation of the program structures in which the operation is different between the old and new language specifications follows the specifications of the C language standard. For example, warning messages are generated for changes in the rules for integer expansion, but no warning messages are generated for three-letter expressions (trigraph sequences) or add escape sequences.
ANSI

The C language specifications used at analysis time—and related headers—are based strictly on the ANSI C language standard. They do not include Fujitsu C compatibility extensions. However, the extended specifications in the Extended ANSI option that do not affect programs that strictly adhere to the C language standard are included; warning messages are generated for some of these extensions. Regarding the identifiers in the standard header, only the identifiers defined in the C language standard are visible.

Other C Options

The following C options can be specified in the Other Options input field:

/A
This option ignores all predefined assertions and macros except those starting with ___ (two underscores)

/D name=token
This option assigns name to the specified tokens as in the #define preprocessing directive. If =token is not specified, a value of 1 is assumed.

/I dir
This option changes the search for include files whose names do not start with \ / or hard-disk drive-name so that the directory specified in dir is searched first and then the directory-installed standard include files are searched.

If multiple directories are specified in multiple /I options the directories are searched in the order specified.

Files specified between the quotation marks (") are searched in the following order:
1. The current directory that contains #include preprocessing directives
2. The directory specified by the /I option
3. The default directory

Files specified between the < sign and the > sign are searched in the following order:
1. The directory specified by the /I option
2. The default directory

When the include file option is specified using an absolute pathname, only the absolute pathname is used for the search.

This option has no effect if a directory is not specified.

/U name
This option renders the definition of name invalid in the same manner as the #undef preprocessing directive. If the same name is specified by both the /D and /U options, the name is not defined, regardless of the order of option specification.
/w
Suppresses the printing of warning messages.

/wc
c:{1|2|3|4|5}
This option detects redundant and nonportable expressions, and possibly bugs in the code. The level of check, c, can be 1, 2, 3, 4 or 5.

- Checking level 1
  Report Unused variables
  ```c
  main() {
    int a ; <--------
  }
  ```
  Report uninitialized variables
  ```c
  main() {
    int a, b ;
    a = b ; <--------
  }
  ```

- Checking level 2
  Include Checking level 1.
  Report return statements if they never return a value in the non-void type function.
  ```c
  int func() {
    int a ;
    a = 1 ;
    return ; <--------
  }
  ```

- Checking level 3
  Include Checking level 2.
  Report static pointer variables which are uninitialized.
  ```c
  main() {
    int a ;
    static int *b = &a ; <--------
    return 0 ;
  }
  ```
Report return statements if they have the address to an auto variable.

```c
int *func() {
    auto int a;
    a = 1;
    return &a;  
}
```

- Checking level 4
- Include Checking level 3.
- Report functions with no type.

```c
func() {  
    int a;
    a = 1;
    return a;
}
```

- Report functions if they do not have prototype declaration.

```c
int test() {
    func();  
    return;
}
```

- Report constant symbols if they are uninitialized.

```c
void func() {
    const int a;  
    return;
}
```

- Checking level 5
- Include Checking level 4.
- Report switch statements without a default label.

```c
void func() {
    int a,b;
    a = 1;
    switch (a) {
    case 1:
        b = 1;
        break;
    case 2:
        b = 2;
        break;
    }
}
Analyzing a Program

Click the Start Analysis command in the Analyze Menu to start the analysis of the selected source files. If the Visual Analyzer detects errors during analysis, the contents of the errors appear in the Analysis Message window.

Note

If there are more than two program units or functions with the same name in the source files, analysis results are unpredictable. For the analysis results data file, see Analysis Results Data File.

Displaying Analyzed Information

When the View Call Graph command in the Analyze Menu is clicked, the Call Graph window is shown. For the Call Graph window, see Call Graph.

Analysis Results Data File

Analysis results data is saved in the directory which is specified in the project directory or option dialog box.

Saving Analysis Results Data

An analysis results data file with an extension of .q is created when you run an analysis. For example, if the source filename is "test.c" an analysis results data file named "test.q" is created. If the contents of a source file were changed after its analysis results data was saved, analysis results are not displayed correctly. If the directory of the created analysis results data file was changed or the created names were changed, the analysis results are not displayed correctly.

Updating an Analysis Results Data File

If previously saved analysis results are already stored in a file and an analysis of the corresponding source program file is performed again, the previous file is replaced with new analysis results.

Displaying Previously Saved Analysis Results

To display the previously saved analysis results without reanalyzing, open the existing project, and click the View Call Graph command in the Analyze Menu.

Deleting Saved Analysis Results Data

To delete the saved results data file, use the Delete Project command or remove the source file using Select Files dialog box. When user uses the Delete Project command, all the analysis results data files are deleted. When user uses the Select Files dialog box, only the analysis results data related the source file is deleted.
Chapter 3 -- Call Graph

This chapter explains how to operate the Call Graph window.

After analyzing the source files, use the View Call Graph command in the Analyze Menu to display the Call Graph window. The Call Graph presents program diagrams with simple and intuitive graphical structures.
Call Graph Display

The figure titled *Example of Call Graph Window* shows an example of the Call Graph window.

**Example of Call Graph Window**

Each box represents a program unit or function.

1. The box representing a Fortran main program is drawn with a thick line.
2. The box for a Fortran subprogram is drawn with a thin line.
3. The box for a Fortran block data subprogram is drawn with a dotted line, and the box for a C function
is drawn with a broken line.

4. The I symbol in the box represents a Fortran internal subprogram.
5. The M symbol in the box represents a Fortran module subprogram.
6. The lines between the boxes represent call relationships between program units and functions.
7. A blue line indicates a call from a higher box in the hierarchy to a lower box.
8. A red line indicates a call from a lower box to a higher box.
9. When a program unit and function make a recursive call (a call to itself), a red rectangle is displayed at the top right corner of the box, as shown in the following figure:

![Recursive Call Display within the Call Graph Window](image)

**Recursive Call Display within the Call Graph Window**

Boxes and lines are not displayed in the following cases:

1. For a program unit or function name when a program unit or function call exists in the source file, but the program unit or function itself was not found in the analyzed source program (such as in the case of a C library or system call). However, you can display these program units and functions by clicking the All Function command in the popup menu in the Call Graph window. See *Displaying Undefined Program Units or Functions*.
2. For an ENTRY name when a subentry name was specified in an ENTRY statement in the Fortran program.
3. For a function name when a function is called by a pointer in a C program.

**Call Graph Popup Menu**

There are two popup menus associated with the Call Graph window:

1. If your mouse pointer is over a box representing a program unit or function you’ll see the popup menu shown in the figure titled *Call Graph Popup Menu(1)* when you press the right mouse button.
2. If your mouse pointer is not over a box representing a program unit or function, you’ll see the popup menu shown in the figure titled *Call Graph Popup Menu(2)* when you press the right mouse button.
Analysis information can be displayed using the commands in these popup menus and through commands invoked from the menu of the main window (see *The Menu of the Main Window of the Visual Analyzer*).

### Call Graph Popup Menus

The following explains each command in the Call Graph popup menu:

**Previous Focus Level Command**

See *Focus Display*.

**Top Focus Level Command**

See *Focus Display*.

**Zoom Command**

See *Zooming Display*.

**Focus (Callee) Command**

See *Focus Display*. 

Focus (Callee and Caller) Command
See Focus Display.

Linkage Command
See Argument Listing.

Logic Flow Command
See Logic Flow.

Fortran standard Check Command
See Fortran Conformance Testing (Program Unit).

Program Statistics Command
See Statistical Listing.

Cross Reference Command
See Cross Reference Listing.

All Function Command
See Displaying Undefined Program Units or Functions.

Find Command
See Find Function.

Save As Bitmap File Command

Use this command to display the Save as dialog box and save the displaying Call Graph as the bitmap file.

Property Command

Use this command to display the Property dialog box. The source file name, line number and the last update of specified program unit or function are displayed in the Property dialog box.
Zooming Display

An enlarged or reduced Call Graph can be displayed. When the Zoom command in the Call Graph popup menu is clicked, the following sub menu appears.

| 200% | 175% | 150% | 125% | 100% | 75% | 50% |

When a reduction or enlargement factor (indicated as a percentage of the original size) is selected, the Call Graph is reduced or enlarged accordingly. If 50% is clicked, the contents in the boxes are not displayed and only the call relationship is displayed.

Focus Function

The call relationships from or around a specific program unit or function shown in the Call Graph can be focused on as follows:

1. Focus on the call relationships from a specific program unit or function.
2. Focus on the call relationships around a specific program unit or function.

To focus on either of these call relationships, move the mouse pointer to the box of that program unit or function, and click the right mouse button. The popup menu shown in the figure titled Call Graph Popup Menu (Focus Selection) appears. If the Focus (Callee) command is clicked on this popup menu, the call relationships described in 1. are displayed. If the Focus (Callee and Caller) command is clicked on this popup menu, the call relationships described in 2. are displayed.
Call Graph Popup Menu (Focus Selection)

If the Focus (Callee) command is clicked, the call relationships from the selected program unit or function are displayed and the color of its name in the corresponding box is changed to green.

The figure titled Example of Call Graph Display by Focus Function Focus (Callee) shows an example of the display that appears when the Focus (callee) command is clicked.
Example of Call Graph Display by Focus Function Focus (Callee)

If the Focus (callee and caller) command is clicked, the call relationships around the selected program unit or function are displayed, and the color of its name in the corresponding box is changed to green. The figure titled Example of Call Graph Display by Focus Function Focus (Callee and Caller) shows an example of the display that appears when the Focus (callee and caller) command is selected for a function.
Example of Call Graph Display by Focus Function Focus (Callee and Caller)

These Focus operations can be repeated any number of times with a combination of Focus (callee) and Focus (callee and caller) commands. The Focus Level at the top left of the Call Graph shows the number of Focus operations performed. The initial value is 0. To return to the previous display, click the Previous Focus Level command in the Call Graph popup menu. To return to the initial display, click the Top Focus Level command in the popup menu.
Displaying Undefined Program Units or Functions

By default, program units or functions which are called from some functions, but which are not defined in the source files (for example, the C libraries, the system calls, or Fortran system subroutines), are not displayed. To display these program units and functions invoke the All Function command in the Call Graph popup menu.

The bottom color of the box of such a program unit or function is yellow.

Check Function

To check the program, use the Check menu in the main window of the Visual Analyzer. See Inspection of a Program.

Global Information Function

You can easily identify where the global data (common blocks in Fortran for example) is used through the Global menu in the main window. See Global Listing.

Find Function

You can search for the name of a program unit or a function in the Call Graph. When the Find command in the View menu in the main window or the Call Graph popup menu is clicked, the Find dialog box shown in the figure titled Call Graph Find Dialog Box appears. Specify the name to be found in the search name input field and click the Find Next button. The color of the box of the specified program unit or function changes to red. These searches are not case-sensitive.
The Find Next button in the Find dialog box works in the same way as the Find Next command in the View menu in the main window.

**Call Graph Find Dialog Box**

**Function Name List**

When the Function Name List command in the View menu is clicked, the Function Name List window shown in the figure titled *Example of Function Name List Window* appears.

**Example of Function Name List Window**

In the Function Name List window, a list of names of the analyzed program units or functions and their corresponding source filenames are displayed.
When you click on the desired name of the program unit or function in this window, the color of the corresponding box of the Call Graph window changes to red.

When you double-click on the name of the program unit, the Logic Flow window appears. The Logic Flow window is described in Logic Flow.

Click the right mouse button to display the following popup menu. Select Save Text File As to save the Function Name List information as a Text file. Select Save CSV File As to save the Function Name List information as a CSV (Comma Separated Values) file.

Function Name List Popup Menu
Chapter 4 -- Call Tree

This chapter explains how to operate the Call Tree window.

After analyzing the source files and displaying the Call Graph window, use the View Call Tree command in the Analyze Menu to display the Call Tree window. The Call Tree presents program diagrams with simple and intuitive graphical structures.
Each box represents a program unit or function.

1. A Fortran program is shown by [F] mark.
2. An C program is shown by [C] mark.
3. A program unit and function made a recursive call is shown by the following mark and <Recursive>.

```
[F] 54: track <Detail>: 4
[F] 56: sensit
[C] 14: sizes_
[C] 15: sizes_ <Recursive>: 14
[F] 16: trace <Detail>: 3
```

**Recursive Call Display within the Call Tree Window**

Boxes and lines are not displayed in the following cases:

1. For a program unit or function name when a program unit or function call exists in the source file, but the program unit or function itself was not found in the analyzed source program (such as in the case of a C library or system call). However, you can display these program units and functions by clicking the All Function command in the popup menu in the Call Tree window. See Displaying Undefined Program Units or Functions.

2. For an ENTRY name when a subentry name was specified in an ENTRY statement in the Fortran program.

3. For a function name when a function is called by a pointer in a C program.

**Call Tree Popup Menu**

There are two popup menus associated with the Call Tree window:

1. If your mouse pointer is over a box representing a program unit or function you’ll see the popup menu shown in the figure titled Call Tree Popup Menu(1) when you press the right mouse button.

2. If your mouse pointer is not over a box representing a program unit or function, you’ll see the popup...
menu shown in the figure titled *Call Tree Popup Menu(2)* when you press the right mouse button.

### Call Tree Popup Menus

The following explains each command in the Call Tree popup menu:

**Logic Flow Command**

See *Logic Flow*.

**Fortran Standard Check Command**

See *Fortran Conformance Testing (Program Unit)*.

**Program Statistics Command**

See *Statistical Listing*.

**Cross Reference Command**

See *Cross Reference Listing*.
Open All Command
Display all the program units and functions.

Close All Command
Display the only top level program units or functions.

Recursive Command
Display the recursive program unit or function.

Detail Command
Display the calling program unit or function specified program unit or function.

All Functions Command
See Displaying Undefined Program Units or Functions.

Find Command
See Find Function.

Save As Text File Command
Use this command to display the Save the Call Tree as Text file.

Find Caller Command
Use this command to display the Find Caller Dialog box.

Linkage Command
See Argument Listing.

Property Command
Use this command to display the Property dialog box. The source file name, line number and the last update of specified program unit or function are displayed in the Property dialog box.
Displaying Undefined Program Units or Functions

By default, program units or functions which are called from some functions, but which are not defined in the source files (for example, the C libraries, the system calls, or Fortran system subroutines), are not displayed. To display these program units and functions invoke the All Function command in the Call Tree popup menu.

The bottom color of the icon of such a program unit or function is yellow.

Find Function

You can search for the name of a program unit or a function in the Call Tree. When the Find command in the View menu in the main window or the Call Tree popup menu is clicked, the Find dialog box shown in the figure titled Call Tree Find Dialog Box appears. Specify the name to be found in the search name input field and click the Find Next button. The color of the box of the specified program unit or function changes to red. These searches are not case-sensitive.

The Find Next button in the Find dialog box works in the same way as the Find Next command in the View menu in the main window.
Chapter 5 -- Logic Flow

This chapter explains how to operate the Logic Flow window, which displays the internal structure of a program unit or function visually.

Call Graph Display

The figure titled *Example of Call Graph Window* shows an example of the Call Graph window.
There are two ways to display the Logic Flow. One is to double-click the name of the program unit or function in the Function Name List window. The other is to click the Logic Flow command in the Call Graph or Call Tree popup menu which appears when the mouse pointer is moved to the box of the program unit or function and the right mouse button is clicked, as shown in the figure below.

![Call Graph View with Logic Flow Selected](image-url)
Logic Flow Popup Menu

When the right mouse button is clicked in the Logic Flow window, the Logic Flow popup menu shown in the figure titled Logic Flow Popup Menu appears.

Comment Command

Use this command to determine whether or not comment lines will be displayed in the Logic Flow window.

User Include Command

Use this command to determine whether or not user-included lines will be displayed in the Logic Flow window.

System Include Command

Use this command to determine whether or not system-included lines will be displayed in the Logic Flow window.

Find Command

See Searching for Names.

Save As Source File Command

Beautified indentation displayed in the Logic Flow windows can now be saved as Fortran source. When you select the Save Source File As command, the following dialog appears.
**Save Source File As Dialog Box**

You must specify the full name including the drive name in the File Name field. Click the Tabs button to change the way tabs are expanded and displayed. Click the Spaces button to change the number of spaces to indent.

**Logic Flow Display**

You can display information about the flow of both Fortran and C program logic.

**Fortran Program Logic Flow Display**

The figure titled *Fortran Program Logic Flow Window Configuration* shows the configuration of the Fortran program Logic Flow window.
Fortran Program Logic Flow Window Configuration

For Fortran programs, only the specified program unit is displayed in one Logic Flow window except for the following two cases.

1. When the internal subprogram is specified, the parent program unit and all its children program units are displayed.
2. When the module subprogram is specified, the whole of the module subprogram is displayed.
3. The (line) column displays line numbers in the source file.
The (inc) column displays the nesting level of an INCLUDE statement and the #include line in the source program. An IF statement, DO statement, CASE statement, GOTO statement, WHERE statement, CYCLE statement and EXIT statement is displayed as shown in the following figure:

IF statement display example

```
if (kind .eq. 1) then
  call report (Rbuf, indata)
else
  stop
endif
```

DO statement display example

```
do 100 L=1,10
  write(6,*) out
100  continue
```

CASE statement display example

```
selectcase (i)
  case (1)
    call test1()
  case default
    call test2()
end select
```

GOTO statement display example

```
goto 999  -> 10
```
CYCLE statement display example

```
do 30 i=1,20
   if(K(i) > 10) then
      write(6,*) "END"
      cycle
   endif
30   continue
```

WHERE statement display example

```
where (a <= 1.0)
   a = a - b
else where
   a = a - 10.0
end where
```

EXIT statement display example

```
do 40 i=1,20
   if(K(i) > 10) then
      write(6,*) "END"
      exit
   endif
40   continue
```

**C Program Logic Flow Display**

The figure titled *C Program Logic Flow Window Configuration* shows the configuration of the C program Logic Flow window.
C Program Logic Flow Window Configuration

For C programs, the whole of the source which contains the specified function is displayed in one Logic Flow window. The (line) column displays the line numbers in the source file. The (inc) column displays the nesting level of an #include line in the source program. An if statement, for statement, switch statement, break statement, goto statement, while statement, return statement, continue statement, structure specifier or union specifier is displayed as shown in the following figure:
if statement display example

```c
if (*p == '1') {
    *p = '2';
}
```

for statement display example

```c
for (i=0; i<10; i++) {
    a[i] = 0;
}
```

switch and break statement display example

```c
switch (i) {
    case 0:
        i = 1;
        break;
    case 1:
        i = 2;
        break;
    default:
        i = 3;
}
```

goto statement display example

```c
goto jump;
```

return statement display example

```c
return (i);
```
while statement display example

```c
while (i < j) {
    i = i + 2 ;
}
```

structure specifier display example

```c
struct s_tag {
    char *indata ;
    long length ;
} s ;
```

for and continue statements display example

```c
for (i=0; i<10; i++) {
    if (a[i] == 0)
        continue ;
    a[0] = 0 ;
}
```

union specifier display example

```c
union u_tag {
    long data1 ;
    long data2 ;
} u ;
```
Searching for Names

A name search can be performed within a source program displayed in the Logic Flow display. See *The Find Command* for details. When the Find command in the View menu in the main window or the Logic Flow popup menu is clicked, the Find dialog box shown in the figure titled *Logic Flow Find Dialog Box* appears.

![Logic Flow Find Dialog Box](image)

### Logic Flow Find Dialog Box

When you first invoke the Find dialog box for the Logic Flow view, the word on which your cursor is currently placed appears in the Find What input field. Accept this as your search string or enter another string, and then click on the Find button to start the search. If the search finds the specified name, the color of the first line containing the specified name changes to green, and the color of the rest of the lines containing the specified name changes to red.

When the Find Next command in the View menu is clicked again, the next searched line’s color changes to green. When the Find Previous command in the View menu is clicked, the previously searched line’s color changes to green.

### Fortran Search Targets

For Fortran programs, when the Find Using Name check box is checked, the target of the search names are variables and procedure names. Comments, character literal constants, characters in the FORMAT statements and Fortran keywords (IF, READ, WRITE, etc.) are not targets.

### C Search Targets

For C programs, when the Find Using Name check box is checked, the targets of the search names are variable names, function names, tag names and macro names. Comments, string literals and reserved words are not targets.
Chapter 6 -- Statistical Listing

This chapter explains the display of statistical information. To display statistical information about any program unit or function in a Call Graph, perform the following actions:

1. Move the mouse pointer to the box of that program unit or function
2. Click the right mouse button.
   The Call Graph popup menu shown in the figure titled Call Graph View with Program Statistics Selected appears.
3. Click the Program Statistics command in the popup menu.
   A Program Statistics window displaying statistical information about that program unit or function appears.
Call Graph View with Program Statistics Selected

Popup Menu of Statistical Listing

When the right mouse button is clicked in the Program Statistics window, the Logic Flow popup menu shown in the figure titled *Popup Menu of Statistical Information* appears.
Popup Menu of Statistical Information

Copy Command

Use this command to copy the specified area to the Windows clipboard. This is the same as the Copy command in the Edit menu in the main window.

Select All Command

Use this command to select the entire contents of the Program Statistics window. This is the same as the Select All command in the Edit menu in the main window.

Logic Flow Command

Use this command to display the Logic Flow window of a given program unit or function.

Find Command

Use this command to display the Find dialog box and search for a character string in the Summary window. See Searching for Character Strings.

Save AS Text File Command

Use this command to save the displaying information as the Text file.

Displaying Statistical Information

This section explains statistical information for Fortran programs and C programs.

Displaying Statistical Information For Fortran Programs

The figure titled Fortran Program Program Statistics Window Configuration shows the configuration of a Fortran program Program Statistics window.
Fortran Program Program Statistics Window Configuration

The Program Statistics window displays the following information:

1. **Program name**
   - The program unit name
2. **Source file**
   - The name of the source file containing the program unit
3. Program type
   The type of program unit (main, subprogram, module, or block data)

4. Complexity
   The program’s complexity. See *Measuring the Complexity of Fortran Programs*

5. Number of statements
   The number of statement lines of in the program unit containing included lines

6. Comment lines
   The number of comment lines

7. Include lines
   The number of included lines

8. Callee
   The callee program unit name. An asterisk(∗) is added in front of the program unit name if the pro-
   gram unit is not included in the same file as the source program being analyzed.

9. Caller
   The caller program unit name

10. OPEN statement
    The number of OPEN statements

11. CLOSE statement
    The number of CLOSE statements

12. READ statement
    The number of READ statements

13. WRITE statement
    The number of WRITE statements

14. CALL statement
    The number of CALL statements

15. ENTRY statement
    The number of ENTRY statements

16. IF statement
    The number of IF statements

17. ELSE IF statement
    The number of ELSE IF statements
18. DO statement
   The number of DO statements
19. DO WHILE statement
   The number of DO WHILE statements
20. DO UNTIL statement
   The number of DO UNTIL statements
21. SELECT CASE statement
   The number of SELECT CASE statements
22. GOTO statement
   The number of GOTO statements
23. ALLOCATE statement
   The number of ALLOCATE statements
24. DEALLOCATE statement
   The number of DEALLOCATE statements
25. ASSIGNMENT statement
   The number of ASSIGNMENT statements

Displaying Statistical Information For C Programs

The figure titled C Program Program Statistics Window Configuration shows the configuration of a C program Program Statistics window.
The Program Statistics window displays the following information:

1. Function name
   - The function name
2. Source file
   - The name of the source file containing the function
3. Storage Class
   The storage class of the function.

4. Complexity
   The program’s complexity. See *Measuring the Complexity of C Programs*.

5. Number of statements
   The number of statements

6. Callee
   The callee function name. An asterisk(*) is added in front of the function name if the function is not included in the same file as the source program being analyzed.

7. Caller
   The caller function name

8. if statement
   The number of if statements

9. switch statement
   The number of switch statements

10. while statement
    The number of while statements

11. do statement
    The number of do statements

12. for statement
    The number of for statements

13. continue statement
    The number of continue statements

14. break statement
    The number of break statements

15. return statement
    The number of return statements

16. goto statement
    The number of goto statements

17. asm statement
    The number of asm statements
18. label
   The number of labels

19. case label
   The number of case labels

20. default label
   The number of default labels

Measuring the Complexity of Fortran Programs

The initial value for the complexity of each program unit is one. Complexity increases according to the following rules:

1. The number of labels specified on unconditional, computed and assigned GOTO statements. GOTO 100 for example counts 1 up to complexity. If the same label appears in a GOTO statement, complexity excludes the number of duplicated labels for example,

   \[
   \text{GOTO (M), 10,20,30 } \quad \text{! Plus 3} \\
   \text{GOTO (M), 10,10,20 } \quad \text{! Plus 2} \\
   \text{GOTO (M), 10,10,10 } \quad \text{! Plus 1}
   \]

2. The number of labels specified on arithmetic IF statements. If the same label appears in an IF statement, complexity excludes the number of duplicated labels, for example,

   \[
   \text{IF (M), 10,20,30 } \quad \text{! Plus 3} \\
   \text{IF (M), 10,10,20 } \quad \text{! Plus 2} \\
   \text{IF (M), 10,10,10 } \quad \text{! Plus 1}
   \]

3. Each appearance of a logical IF, block IF or ELSE statement.

4. Each appearance of a DO, DO UNTIL, or DO WHILE statements.

5. Each CASE appearance in a SELECT CASE statement.

6. The number of labels with the END=, ERR= or EOR= specifiers on I/O statements. If the same label appears in an I/O statement, complexity excludes the number of duplicated labels, for example:

   \[
   \text{READ (10,*,EOR=10,END=20,ERR=30) } \quad \text{! Plus 3} \\
   \text{READ (10,*,EOR=10,END=10,ERR=20) } \quad \text{! Plus 2} \\
   \text{READ (10,*,EOR=10,END=10,ERR=10) } \quad \text{! Plus 1}
   \]

Measuring the Complexity of C Programs

The initial value for the complexity of each function is one. Complexity increases according to the following rules:

1. Complexity increases by one for each appearance of an if, while, for, do, goto, or switch statement.
2. Complexity increases by one for each appearance of CASE LABEL or STATEMENT LABEL.

**Interpreting Complexity Values**

A high complexity value indicates that the program unit or function has a complex control structure, increasing the likelihood that it contains bugs.

**Searching for Character Strings**

See the discussion of *The Find Command* for details on searching for character strings.
Chapter 7 -- Argument Listing

This chapter explains the display of Argument Information. Use the following procedure to display or check the argument information passed between any of the program units or functions shown in the Call Graph.

1. Move the mouse pointer to the box of the desired program unit or function.
2. Click the right mouse button.
   The popup menu shown in the figure titled *Call Graph View with Linkage Selected* appears.
3. Click the Linkage command in this popup menu.
   The mouse pointer changes to a hand symbol.
4. Move the hand-shaped pointer to the box of a program unit or function that calls or is called from the specified program unit or function.
   Click the right mouse button.
   A Linkage window displaying information on arguments passed between the two program units or functions appears.
   If a Fortran program and a C program are linked together, the argument information is not displayed, and a message is displayed.
When the right mouse button is clicked in the Linkage window, the Logic Flow popup menu shown in the figure titled *Linkage Listing Popup Menu* appears.
Copy Command

Use this command to copy the specified area to the Windows clipboard. This is the same as the Copy command in the Edit menu in the main window.

Select All Command

Use this command to select all the information areas in the Linkage window. This is the same as the Select All command in the Edit menu in the main window.

Logic Flow Command

Use this command to display the Logic Flow window for the program unit or function that contains the specified actual argument.

Find Command

Use this command to display the Find dialog box and search for a character string in the information that is displayed in the Linkage window. See Searching for Character Strings.

Save As Text File Command

Use this command to save the displaying information as the Text File.

Displaying Argument Listing

This section explains argument information for Fortran programs and C programs.

Displaying Argument Listing for Fortran Programs

The figure titled Example of Fortran Program Argument Information Display shows an example of the Linkage window display for a Fortran program.
Example of Fortran Program Argument Information Display

The dummy argument information contains the following:

1. No
   The order of the specified dummy argument

2. Name
   The dummy argument name. Names are displayed entirely in lowercase alphabetic characters.
3. Kind
   The kind of dummy argument

4. Type
   The type of dummy argument

5. Line Number
   The line number of declarations, definitions, assignments or references of dummy arguments. Definition, declaration, assignment and reference are classified as follows:
   a. (Declaration) Declaration
   b. (Definition) Definition or assignment
   c. (Reference) Reference

The actual argument information displays the following:

1. The source filename and line number calling the program unit.
   The message generated when an integrity check (the number of arguments, type) between actual arguments and dummy arguments returns an abnormal result.

2. No
   The order of the specified actual argument.

3. Name
   The actual argument name. Names are displayed entirely in lowercase alphabetic characters. If the actual argument, named constant, or statement number is specified, N/A is displayed. If an element of structure type is specified, only the variable name of structure type is displayed.

4. Kind
   The kind of actual argument

5. Type
   The type of actual argument

The file listing displays the source or include filenames with an identification number. For example:

\begin{verbatim}
D:\users\demo\test.c
[#1]D:\FF95\include\stdio.h
\end{verbatim}

The source filename and the names of included files are included, along with an identification number for included files. When the line number where the program unit name or parameter name appears is a line number of the included file, this identification number is displayed just behind the line number. If the file identification number is not specified, it is the line number of the source file.
Displaying Argument Listing for C Programs

The figure titled Example of C Program Argument Information Display shows an example of the Linkage window display for a C program.
The parameter information displays the following:

1. No
   The order of the specified parameters.
   For functions, "-" (rather than the order number) is listed.

2. Name
   The parameter name or function name

3. Type
   The parameter type or function result type

4. Line Number
   The line number of declarations, definitions, assignments or references of parameters. Definition,
   declaration, assignment and reference are classified as follows:
   a. (Declaration) Declaration
   b. (Definition) Definition or assignment
   c. (Reference) Reference

The argument information displays the following:

1. The source filename and line number calling the function.

2. The message generated when an integrity check (the number of arguments, type) between an argu-
   ment, parameter and function result returns an abnormal result.

3. The type of function result and order and type of argument specified.

The file information displays the following:

The source filename and the names of included files are included, along with an identification number for
included files. When the line number where the function or parameter name appears is a line number of
the included file, this identification number is displayed just behind the line number. If the file identification
number is not specified, it is the line number of the source file.

Searching for Character Strings

See the discussion of The Find Command for details on searching for character strings.
Chapter 8 -- Cross-Reference Listing

This chapter explains the display of Cross-Reference Listing. To display Cross-Reference information about any program unit or function in the Call Graph, perform the following actions.

1. Move the mouse pointer to the box of that program unit or function
2. Click the right mouse button.

   The popup menu shown in the figure titled Call Graph View with Cross Reference Selected appears.

Click the Cross Reference command in this popup menu to display Cross-Reference information about the selected program unit or function.
Cross-Reference Listing Popup Menu

When the right mouse button is clicked in the Cross Reference window, the Cross-Reference popup menu shown in the figure titled Cross-Reference Listing Popup Menu for Fortran (for Fortran programs) or the figure titled Cross-Reference Listing Popup Menu for C (for C programs) appears.
For Fortran Programs

Variable/Procedure command

Use this command to determine whether or not variable and procedure information will be displayed.
Derived Type Command
Use this command to determine whether or not derived type information will be displayed.

Macro Command
Use this command to determine whether or not macro information will be displayed.

Statement Label Command
Use this command to determine whether or not statement label information will be displayed.

File Command
Use this command to determine whether or not filename information will be displayed.

For C Programs

Variable/Function Command
Use this command to determine whether or not variable and function information will be displayed.

Tag/Typedef Command
Use this command to determine whether or not tag and macro information will be displayed.

Macro Command
Use this command to determine whether or not macro information will be displayed.

File Command
Use this command to determine whether or not filename information will be displayed.

Common Commands for Fortran Programs and C Programs

Copy Command
Use this command to copy the specified area to the Windows clipboard. This is the same as the Copy command in the Edit menu.

Select All Command
Use this command to select all the information areas in the active window. This is the same as the Select All command in the Edit menu.

Logic Flow Command
Use this command to display the Logic Flow window for the program unit or function.
Find Command

Use this command to display the Find dialog box and search for a character string in the information that is displayed in the Cross-Reference window. See Character Strings.

Save As Text File Command

Use this command to save the displaying information as the Text file.

Displaying Cross-Reference Information

This section explains Cross-Reference information for Fortran programs and C programs.

Displaying Cross-Reference Information for Fortran Programs

For Fortran programs, cross-reference information comprises:

1. The variable end procedure name listing
   The variable name and procedure name listing displays cross-reference information about variable names procedure names and so on.

2. The derived type name listing
   The derived type name listing displays cross-reference information about derived type names.

3. The macro name listing
   The macro name listing displays information about the macro definitions or undefinitions.

4. The statement label listing
   The statement label listing displays Cross-Reference information about statement labels. The figure titled Example of a Fortran Program Cross-Reference Window shows an example of a Fortran program Cross-Reference window.

5. The filename listing.
Example of a Fortran Program Cross-Reference Window

The variable and procedure name listing displays the following information:

1. **Name**
   - The name used. All names are displayed in lowercase alphabetic characters.

2. **Attribute**
   - The attribute of the name
3. Type
   The type of the name. N/A (for Not Available) is displayed if there is no type (subroutine, etc.)

4. Line Number
   The line number where variables and procedures are defined, declared, assigned or referenced.

5. Line Number
   The line number of declarations, definitions, assignments or references of variables or procedure names. Definition, declaration, assignment and reference are classified as follows:
   a. (Declaration) Declaration
   b. (Definition) Definition or setting
   c. (Reference) Reference

The derived type name listing displays the following information:

1. Type Name
   The type name of the derived type

2. Definition
   The line number of the statement which defines the derived type name

3. Line Number
   The line numbers by which the derived type name is referenced

The macro name listing displays the following information:

1. Macro Name
   The macro name used

2. Replacement
   The replacement character is displayed at the macro definition. When the macro definition is invalidated, this column becomes a blank space.

3. #define
   The defined line number is displayed at the macro definition. When the macro definition is invalidated, this column becomes a blank space.

4. #undef
   When the macro definition is validated, the invalidated line number is displayed. This column becomes a blank space at the macro definition.

The statement label listing displays the following information:

1. Label
   The label appears in the source program
2. Kind
   The kind of the statement label

3. Definition
   The line number of the statement which defines the statement label

4. Reference
   The line numbers by which the statement label is referenced

The filename listing displays the following information:

The source filename, including variable names, procedure names, derived type names, macro definitions or statement labels and the name of header files included. When the line number where the variable name, procedure name, derived type name, macro definition, or statement label appears is a line number of the included file, this identification number is displayed just behind the line number. If the file identification number is not specified, it is the line number of the source file.

Displaying Cross-Reference Information for C Programs

For C programs, cross-reference information comprises the variable name and function name listing, the tag name listing, the macro listing, and the filename listing. The variable and function name listing displays cross-reference information about variable names, function names, parameter names, and enumerator names. The tag listing displays information about the tag declarations of structure types, union types, and enumerated type and the type declarations. Tag/typedef listings display information about tag declarations of structure, union, and enum and typedef. The macro name listing contains macro definitions.

The figure titled Example of a C Program Cross-Reference Window shows the configuration of a C program Cross-Reference window.
Example of a C Program Cross-Reference Window

The variable name and function name listing displays the following information:

1. Name
   The variable name, the function name, the parameter name, or the enumerator name

2. Attribute
   One of the following attributes is displayed.
   a. variable(variable name)
b. function(function name)
c. parameter(parameter name)
d. enumerator(enumerator name)

3. If the external declaration is declared to the variable or the function, the word "extern" is also displayed in those attributes. If the static declaration is declared to the variable or function, the word "static" is also displayed in them. If the prototype declaration exists, the word "prototype" is also displayed in them. If the function declaration exists, the word "definition" is also displayed in them. If the implicit typing exists, the word "implicit" is also displayed in them.

4. Type
   The variable type, the function result type, or the parameter type

5. Line Number
   The line number for definition, declaration, setting, or reference of variable, function, parameter or enumerator name, and each usage. Usage is one of the following:
   a. (Declaration) Declaration
   b. (Definition) Definition or Setting
   c. (Reference) Reference

The tag name listing displays the following information:

1. Name
   The tag name of the structure, union, or enumerated type, or the typedef declaration name

2. Name Kind
   The following kind of the tag name or the typedef declaration name is displayed:
   a. structure tag (structure tag name)
   b. union tag (union tag name)
   c. enumeration (enumeration name)
   d. typedef (typedef name)

3. Type
   The type of the tag name or the typedef declaration name

4. Definition
   The defined line number

The macro name listing displays the following information:

1. Macro Name
   The macro name used
2. Replacement
   The replacement character is displayed at the macro definition. When the macro definition is invalidated, this column becomes a blank space.

3. \#define
   The defined line number is displayed at the macro definition. When the macro definition is invalidated, the column becomes a blank space.

4. \#undef
   When the macro definition is invalidated, the invalidated line number is displayed. This column becomes a blank space at the macro definition.

The filename listing displays the following information:

The source filename, including variable names, function names, tag name definitions, or macro definitions and the name of header files included. When the line number where the variable name, function name, tag name definition, or macro definition appears is a line number of the included file, this identification number is displayed just behind the line number. If the identification number is not added to the line number, it is considered the filename.

See the discussion of The Find Command for details on searching for character strings.
Chapter 9 -- Global Listing

This chapter explains the display of Global information. When the Global menu in the main window is clicked, you'll see the following:

Global Menu

Global (C) Command

Use this command to display the global variables in C programs and the function names which use those variables. See Displaying C Global Variables.

Common Definition Command

Use this command to display, for each common block, the common block size, the common block element names, the element types and the Fortran program names using that common block. See Displaying Fortran Common Blocks.

Common Reference Command

Use this command to display, for each Fortran program, the program name and the common block names which are used in that program. See Displaying Fortran Common Blocks.

Module Definition Command

Use this command to display, for each module subprogram, the Fortran module subprogram name and the program names which use that module subprogram. See Displaying Fortran Module Subprogram Relationships.

Module Reference Command

Use this command to display, for each Fortran program, the program name and the module subprogram names which use by that program. See Displaying Fortran Module Subprogram Relationships.
Global Listing Popup Menu

When the right mouse button is clicked in each global information window, the popup menu shown in the figure titled Global Listing Popup Menu appears.

- **Copy Command**
  Use this command to copy the specified area to the Windows clipboard. This is the same as the Copy command in the Edit menu.

- **Select All Command**
  Use this command to select all the information areas in the active window. This is the same as the Select All command in the Edit menu.

- **Find Command**
  Use this command to display the Find dialog box and search for a character string in the listing that is displayed in each Global information window. See Searching for Character Strings.

- **Save As Text File Command**
  Use this command to save the displaying information as the Text file.

Displaying C Global Variables

When the Global (C) command in the Global menu is clicked, the variables in C programs and function names which use those variables are displayed. If no global variable is used in any C program, a message to this effect is displayed. If only Fortran programs are analyzed, the Global (C) command cannot be clicked. The figure titled Example of the C Global Variable Window shows an example of the C Global Variable window.
Example of the Common Definition Window

Displaying Fortran Common Blocks

When the Common Definition command in the Global menu is clicked, the common block size, the common block element names, the element type and the program unit names using this common block are displayed for each common block. The blank common is shown as %blank%. If no common block exists in any of the program units, a message to this effect is displayed. If only C programs are analyzed, the Common Definition command cannot be clicked. The figure titled Example of the Common Definition Window shows an example of the Common Definition window.
Example of the Common Definition Window

When the Common Reference command in the Global menu is clicked, the program name and block names used in that program are displayed for each program. The blank common is shown as %blank%. If no common block exists in any of the program units, a message to that effect is displayed. If only C programs are analyzed, the Common Reference command cannot be clicked. The figure titled Example of the Common Reference Window shows an example of the Common Reference window.
Example of the Common Reference Window

### Displaying Fortran Module Subprogram Relationships

When the Module Definition command in the Global menu is clicked, the module program name and the program names which use that module subprogram are displayed for each module subprogram.

If no module subprogram exists in any of the program units, a message to that effect is displayed.

If only C programs are analyzed, the Module Definition command cannot be clicked. The figure titled *Example of the Module Definition Window* shows an example of the Module Definition window.
Example of the Module Definition Window

When the Module Reference command in the Global menu is clicked, the program name and the module subprogram names which are used in that program are displayed for each program. The variable names in the module subprograms and corresponding local names are also displayed.

If no module subprogram exists in any of the program units, a message to that effect is displayed.

If only C programs are analyzed, the Module Reference command cannot be clicked. The figure titled Example of the Reference Window shows an example of the Module Reference window.
Example of the Reference Window

Searching for Character Strings

See the discussion of *The Find Command* for details on searching for character strings.
Chapter 10 -- Inspection of a Program

This chapter explains the inspection of a program. When the Check menu in the main window is clicked, you'll see the following:

**Check Menu**

**Argument Command**

The argument check is performed for all the program units and functions. See *Checking Inconsistency Between Actual Arguments and Dummy Arguments*.

**Common Block Command**

The common size block check is performed for all common blocks.

**Fortran Standard Command**

Fortran programs are checked to determine if they are compliant with the Fortran 95 standard.

**Inspection of a Program Window Popup Menu**

When the right mouse button is clicked in each inspection of a program window, the popup menu shown in the figure titled *Inspection of a Program Popup Menu* appears.
Copy Command

Use this command to copy the specified area to the Windows clipboard. This is the same as the Copy command in the Edit menu.

Select All Command

Use this command to select all the information areas in the active window. This is the same as the Select All command in the Edit menu.

Find Command

Use this command to display the Find dialog box and search for a character string in the information that is displayed in each inspection of a program window. See Searching for Character Strings. However, the Logic Flow command cannot be used.

Save As Text File Command

Use this command to save the displaying information as the Text file.

Checking for Inconsistencies Between Actual Arguments and Dummy Arguments

When the Argument command in the Check menu is clicked, the argument check (the argument type, the argument kind, the number of arguments, and the function return value) is performed for all the program units and functions. If they are inconsistent, the color of the inconsistencies changes to red, and the Linkage (all) window appears, showing the details of the inconsistency. Otherwise, a message is displayed stating that there is no inconsistency.

Checking the Size of Fortran Common Blocks

When the Common Block command in the Check menu is clicked, the common size check is performed for all common blocks. In this check, the boundary alignment is not considered. If the same common block is declared in more than one program unit and these common block sizes are different, the color of the boxes of the program units which contain these common blocks becomes red, and the Common Block window displays the details of the errors. If the common block does not exist or no error is detected, a message to this effect is displayed. If only C programs are analyzed, the Common Block command cannot be clicked.

Fortran Conformance Testing (All)

When the Fortran Standard command in the Check menu is clicked, the Fortran programs are checked to determine if they are compliant with the Fortran 95 standard. If noncompliant programs are found, the box color of these programs becomes red and the Fortran Standard (All) window which displays the de-
tails appears. If all of the Fortran programs are compliant with the Fortran 95 standard, a message to this
effect is displayed. If only C programs are analyzed, the Fortran Standard button cannot be clicked. When
checking a specific program unit, see *Fortran Conformance Testing (Program Unit)*.

**Fortran Conformance Testing (Program Unit)**

This section explains the method of performing Fortran 95 conformance testing for each Fortran program
unit. When the mouse pointer is moved to the box of each program of Fortran to inspect the Call Graph
and the right mouse button is clicked, the Call Graph popup menu in the figure titled *Call Graph Popup
Menu (Fortran Standard Check Selection)* is displayed. The Fortran conformance testing is performed
when the Fortran Standard Check command is clicked and the result is displayed. When each program
unit within the specified Fortran conforms to the Fortran 95 standard, a message to this effect is displayed.
The Fortran Standard Check in the Call Graph popup menu cannot be used with C programs.
C Program Inspection

You can attempt to detect redundant and nonportable expressions, and possibly bugs in C programs. For more information, see the /wc option.
Searching for Character Strings

See the discussion of *The Find Command* for details on searching for character strings.
Chapter 11 -- Printing

This chapter explains how to print analyzed results using the Print, Print Preview, or Printer Setup command in the File menu in the main window. Each command is explained below.

Print Command

Use this command to display the Print dialog box and print the information in the active window.

Print Preview Command

Shows how the image in the preview window will look when printed.

Printer Setup Command

Use this command to set printer information.

Printing the Call Graph

When the Call Graph window is active and the Print command is clicked, the Call Graph that is shown in the Call Graph window can be printed. You can confirm whether the Logic Flow will fit on one page or not by using the Print Preview command. If the Logic Flow does not fit on one page, it is formatted and printed on multiple pages automatically.

Printing Information Other Than the Call Graph

When a window of information is active and the Print command is clicked, the information will be printed.

Note

Information in the Function Name List window cannot be printed.