

## Relation Analyzer

Welcome to CnPack Relation Analyzer in CnPack IDE Wizards.

This Tool is Used to analyze executable files generated by Delphi, get its packages and units information. And it also can get the relations between packages, and relations between packages and its units.

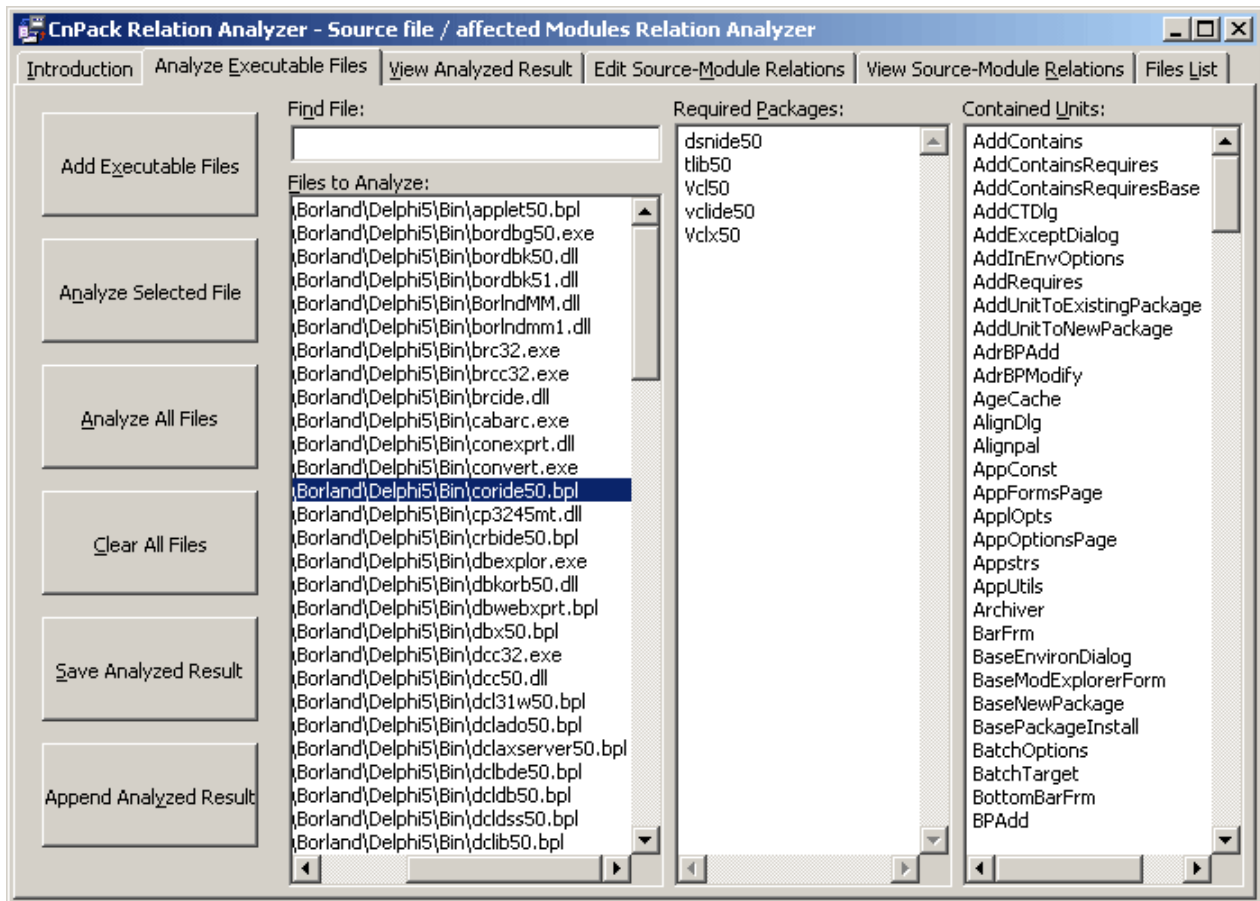
Analyzed result can be saved to an ARF File. And we can create an SMR file from an ARF file to query/view the relations between the executables and source files later.

For common situation, you can open one or more executable files in "Analyze Executable Files" page and analyze them. analyzed result can be save to an ARF file and re-opened in "View Analyzed Result Files" page.

ARF is generated by analyzed result, which only contains the relations between the executables and pascal source file NAME, NO path information and other file types. For advanced situation, you can open an ARF file in "Edit Source-Module Relations" page to edit it and generate an SMR file, which contains the relations between the executables and full path information of source files. Then you can view SMR files in "View Source-Module Relations" page later.

### Analyze Executable Files

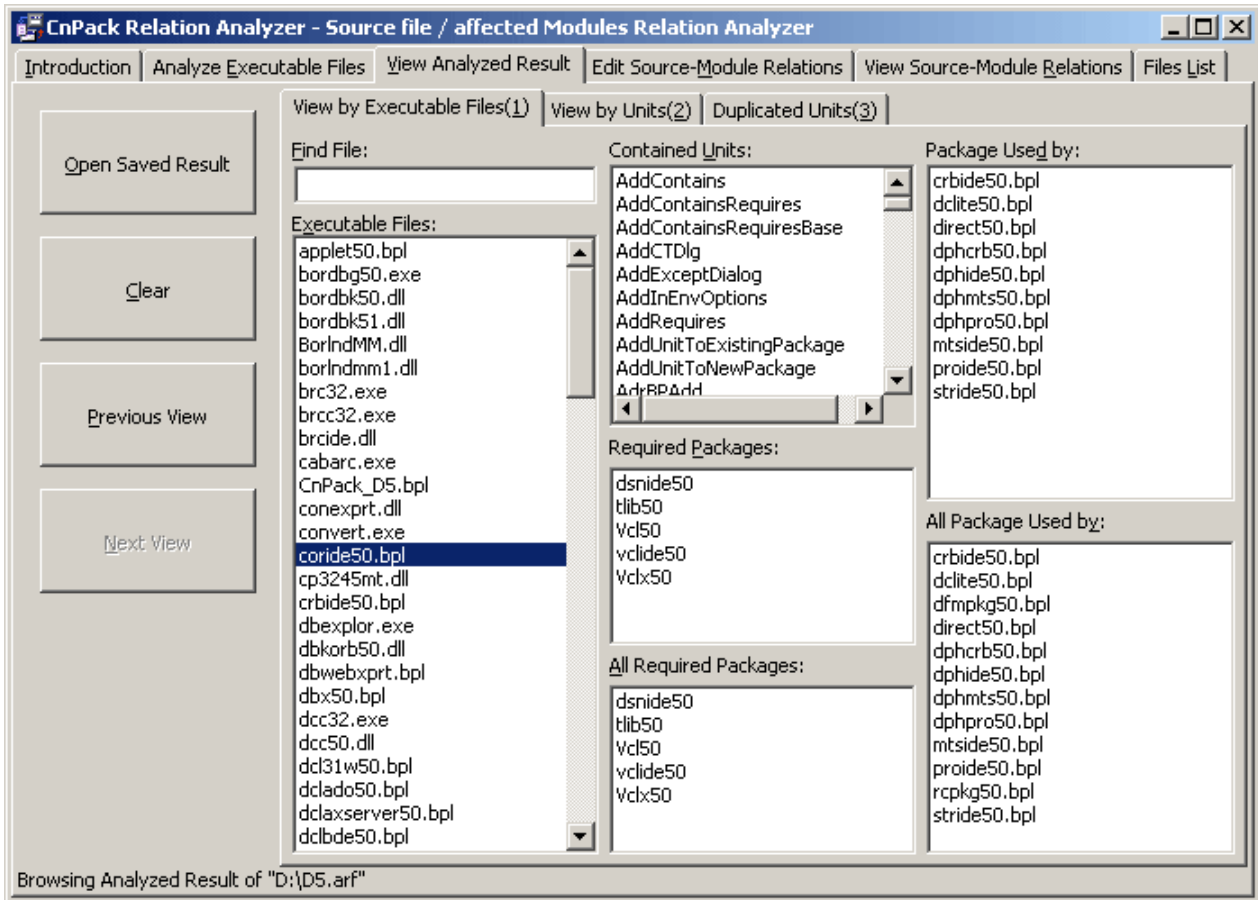
"Analyze Executable Files" is the basic function of Relation Analyzer. It can open executable files and analyze it to get the names of packages and units in executable files. It's shown as below:



- 1 **Add Executable Files:** Open executable files and add to list to analyze.
- 1 **Analyze Selected File:** Analyze the selected file in the list to get its package and unit information.
- 1 **Analyze All Files:** Analyze all files in the list to get their package and unit information
- 1 **Clear All Files:** Clear the file list, include the analyze result.
- 1 **Save Analyzed Result:** Save current analyzed result to an ARF file. It's an INI file, which can be opened by notepad.
- 1 **Append Analyzed Result:** Append current analyzed result to an existing ARF file.

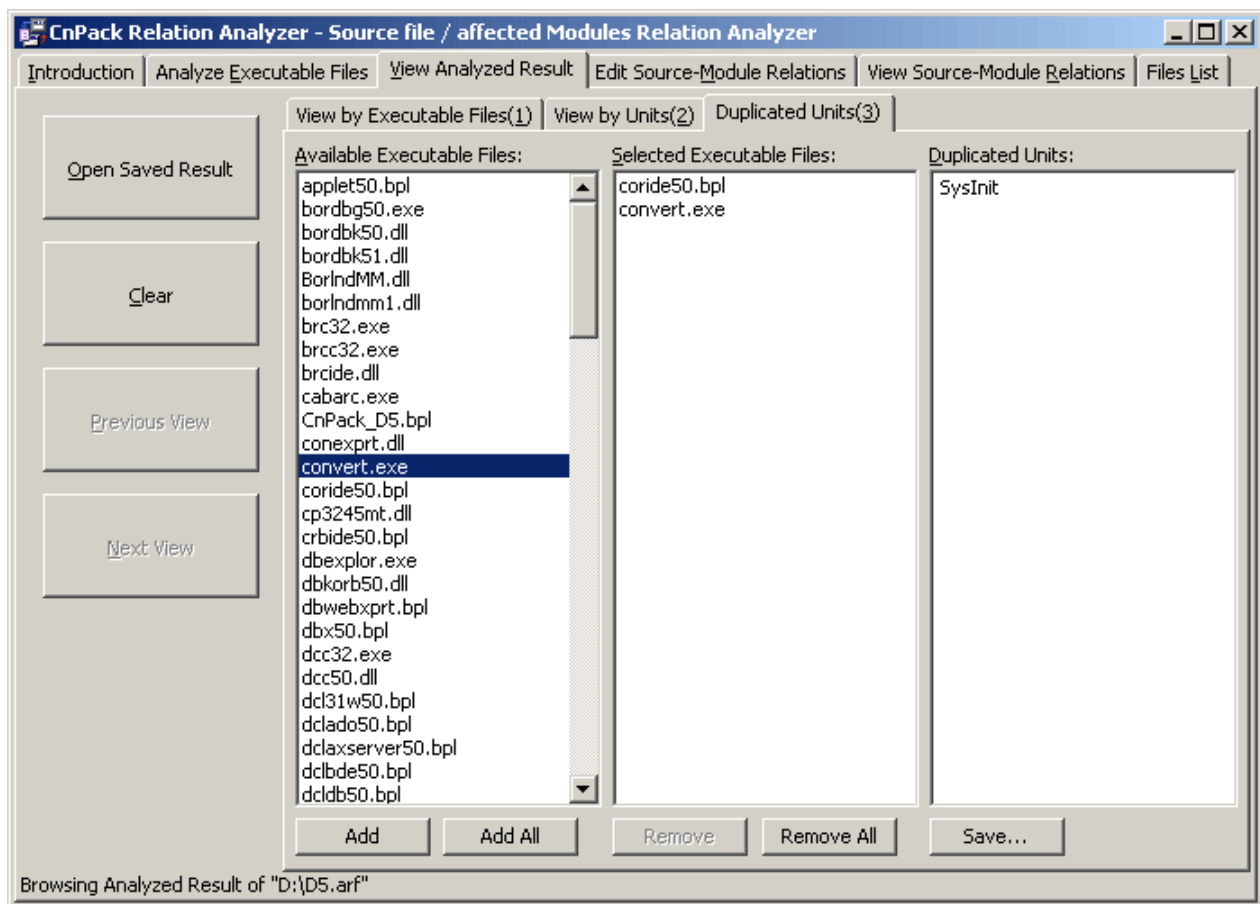
### View Analyzed Result

A saved ARF file can be re-opened here to view its content. This page can view the file relations by executable file or unit. It's shown as below:



- i **Open Saved Result:** Open a saved ARF file and display its content.
- i **Clear:** Clear contents in current page.
- i **View by Executable Files:** Above is shown as the ARF file is viewed by executable files. The executable files are displayed in list. If an executable file is selected, the packages and units which it contains, and the executable files which used it, will be displayed.
  - i **Contained Units:** List the unit names used by selected executable file.
  - i **Required Packages:** List the package names DIRECTLY used(build with, or keyword "requires") by selected executable file.
  - i **All Required Packages:** List the package names directly used(build with, or keyword "requires") and indirectly, recursively used by selected executable file. For example, if a.exe used b.bpl, and b.bpl requires c.bpl, and when a.exe selected, "Required Packages" should contain b.bpl, and "All Required Packages" should contain b.bpl and c.bpl. Generally, "All Required Packages" shows the all required packages to compile this executable file.
  - i **Package Used by:** List the executable files that DIRECTLY used selected executable file(bpl).
  - i **All Package Used by:** List the executable files that directly and indirectly, recursively used selected executable file(bpl). For example, if c.exe build with b.bpl, and b.bpl requires a.bpl, when a.bpl is selected, "Package Used by" should contain b.bpl, "All Package Used by" should contain b.bpl and c.exe.
- i **Previous View:** If double click an executable file in these lists, it will jump to this file and show its relations. Previous View can jump back to see the file just shown, not the previous file in the list.
- i **Next View:** If pressed "Previous View" to jump back, press "Next View" can jump forward.
- i **Find File:** This Edit is used to locate file quickly, wild char supported. Up and down arrow key can move up and down in file list. Ctrl + Home and Ctrl + End can move to the first and last one in file list. Ctrl + Up and Ctrl + Down can find previous one and next one in file list. This function is similar to the one at other pages.
- i **View by Units:** Double clicking a Unit in "Contained Unit" can View this unit's information. The packages used this unit will be displayed as below:

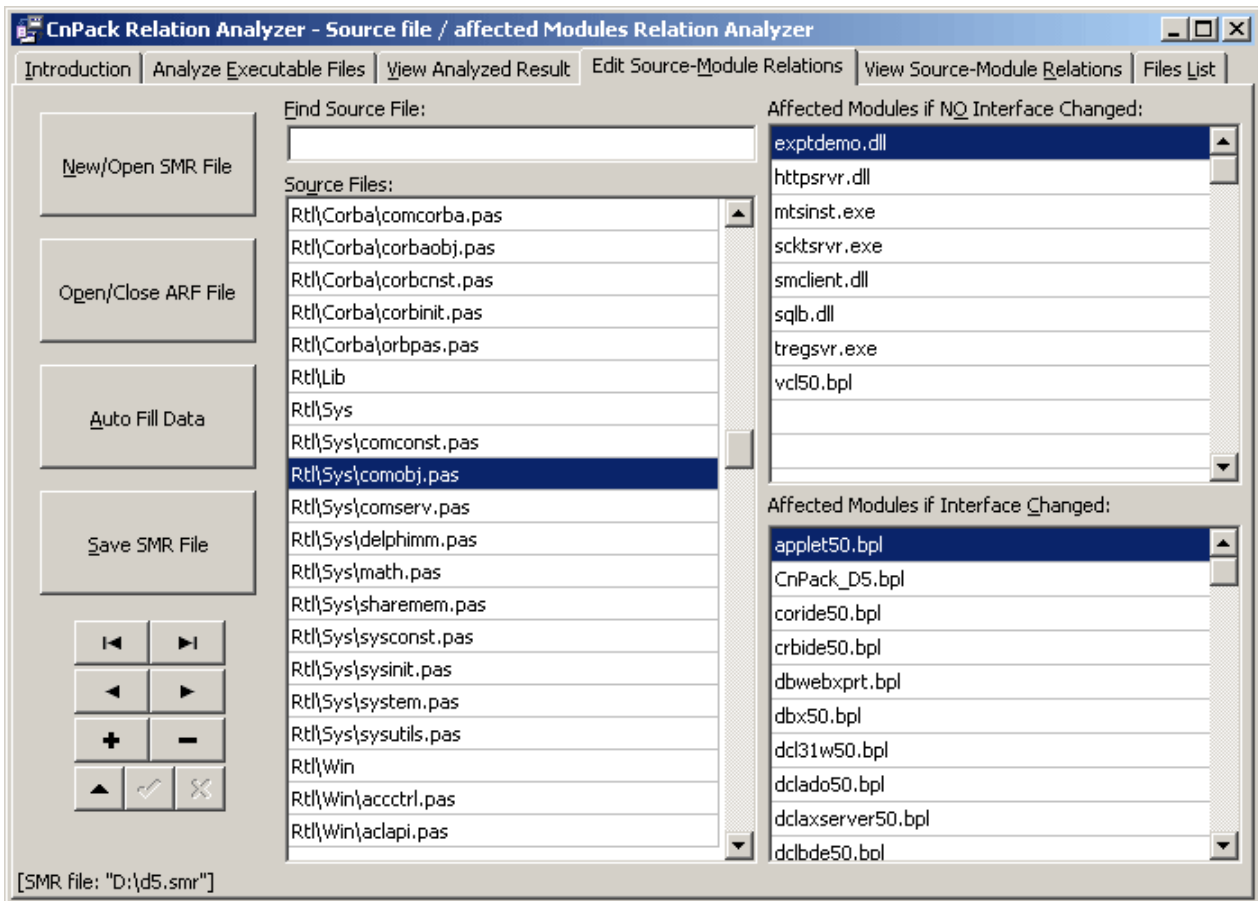




- | **Available Executable Files:** List all executable files in this ARF file.
- | **Add:** Pressing "Add" button or double clicking an executable file can add the file to "Selected Executable Files" List.
- | **Add All:** Pressing "Add All" button can add all files to "Selected Executable Files" List.
- | **Selected Executable Files:** Selected Executable Files List. Duplicated units will be fetched out from them.
- | **Remove:** Remove selected file from "Selected Executable Files" List.
- | **Remove All:** Remove all files from "Selected Executable Files" List.
- | **Duplicated Units:** When "Selected Executable Files" List changed, their units with same name will be displayed in this list.
- | **Save:** Save Duplicated units name to a file.

### Edit Source-Module Relations

An ARF file only contents the relations between executable files and Packages/Units file NAME without the path information. If we want to get and maintain a source file-executable file relation database from a project, we can construct an SMR file, filling the package/unit names with full path information. This page is shown as below:



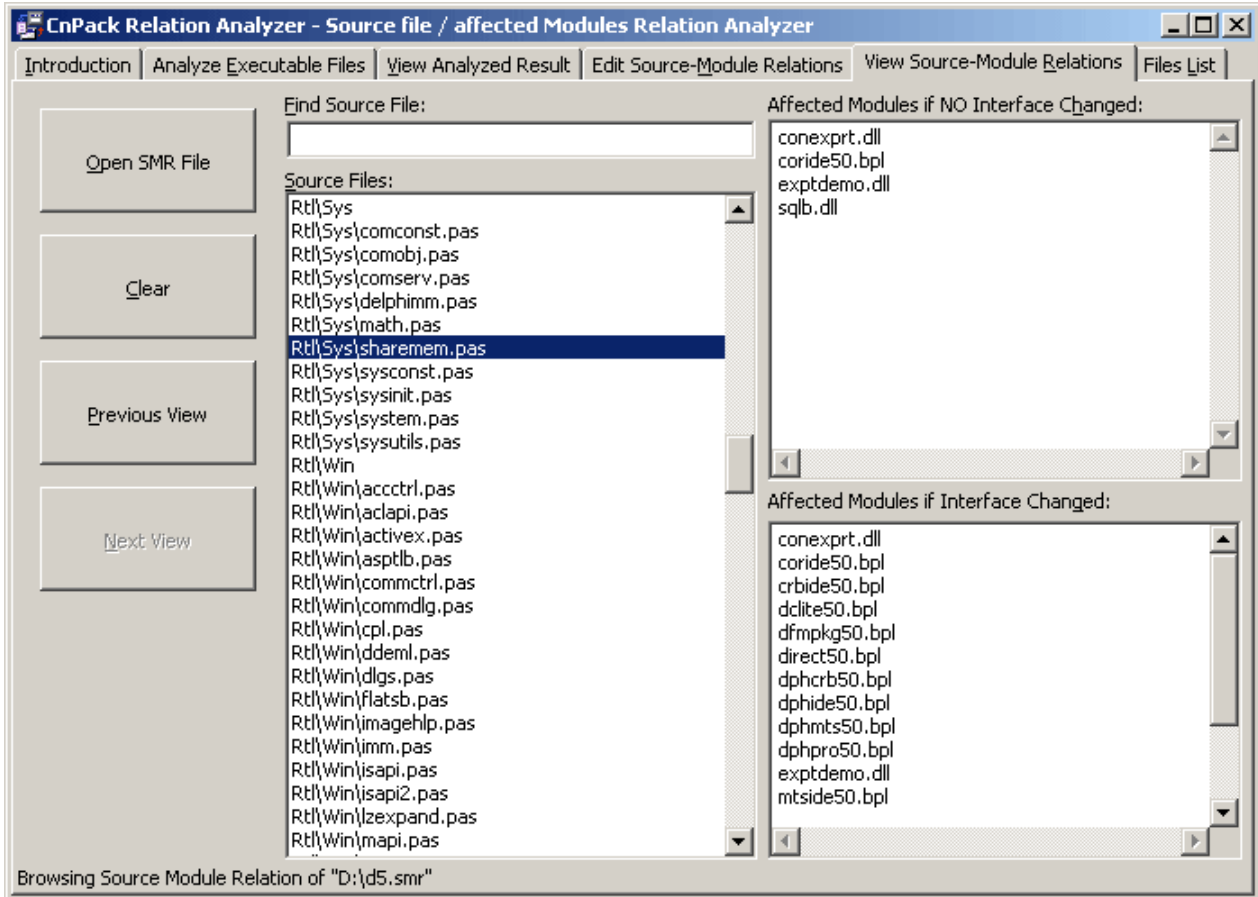
An SMR file contains some source files with full path information and the relations between them and some executable files. To generate a full SMR file, we need an ARF file and a file list. ARF file comes from previous pages, and the file list contains the full path and file names under certain directory, which can be generated from "Files List" Page.

- New/Open SMR File:** This button contains a dropdown menu related to SMR file operations.
  - Open an SMR File:** Open an existing SMR file to edit.
  - Close SMR File:** If an SMR file is opened for editing, this item can close this SMR file. If need saving, please press "Save SMR File" before closing.
  - New an SMR File:** New an empty SMR file for editing.
  - New an SMR File from Files List:** Open a files list file and new an SMR file based on this list file. This function is mostly used when creating SMR file to avoid enter source file path/names manually.
- Open/Close ARF File:** This button contains a dropdown menu related to ARF file operations.
  - Open an ARF File:** Open an existing ARF file for filling SMR file.
  - Close ARF File:** If an ARF file is opened for filling, this item can close this ARF file.
- Auto Fill Data:** If the source files list in SMR are ready, and corresponding ARF file is opened, this function will fill the relations between source files and executable files according to the package/unit name in ARF. For each line in SMR source file, its file name will be extracted and this function will search the same file name in ARF. If matched, the relation is constructed. Note: if there're some files with same name but different path, the filling maybe cause mistake for ARF does not contain the path information. For this situation, we need to adjust them manually.
  - Fill Selected Source-Module Relation:** If a file is selected in "Source File" list, it will search its file name in ARF and get its corresponding executable files' information.
  - Fill All Source-Module Relations:** Search the file name of "Source File" list one by one in ARF file to get their corresponding executable files' information.
- Save SMR File:** If the relations filled, this SMR file, which contains the relations between executable files and certain source files with full path information, can be saved to disk and re-opened later.
- DBNavigator Buttons:** The source files and relations are stored in Data set mode in this program, so there is an DBNavigator to operate the records. Such as new a source file, delete a source file, etc.
- Affected Modules if NO Interface Changed:** If a source file is selected, this list will contain those executable files that use this unit DIRECTLY. In another word, if the implementation part of this unit is modified, the executable files in the list, and only the executable files in the list, need to be re-compiled.

- Affected Modules if Interface Changed:** If a source file is selected, this list will contain those executable files that use this unit directly and indirectly but not recursively. For an example, if a.bpl contains units a.pas, and b.bpl requires a.bpl, c.bpl requires b.bpl. if implementation of a.pas changed, only a.bpl need to re-build, for a.bpl's external interface remains unchanged. But if interface of a.pas changed, a.bpl's external interface will change, too. So b.bpl, which requires a.bpl, need to re-build. And c.bpl does not need re-build for b.bpl's external interface not changed.

### View Source-Module Relations

This page can open an existing SMR file and view it. The page is similar as "Edit Source-Module Relations" shown as below:

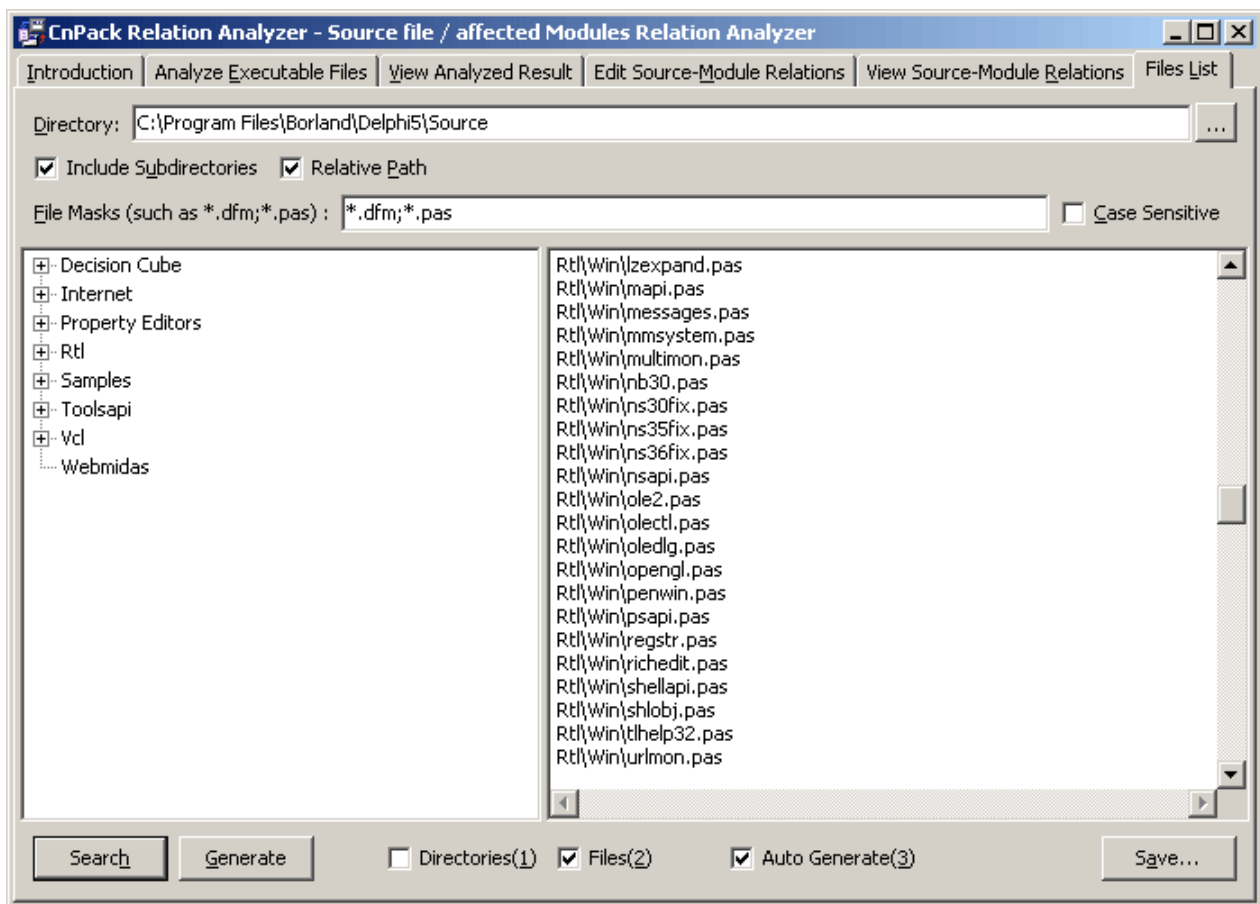


- Open SMR File:** Open an SMR File to view it.
- Clear:** Clear display content.
- Previous View:** Go back to the file just viewed.
- Next View:** Go forward after go back to the file just viewed.

Other UI items are similar to those in "Edit Source-Module Relations".

### Files List

This function is used to generate a source files list under a certain directory for SMR file editing. The page is shown as below. Left is the search result shown as a tree. Right is the file list generated from the tree. The list needs to save to a text file for SMR file editing.



- | **Directory:** Browse and select a directory as a root for searching.
- | **Include Subdirectories:** Whether include subdirectories when searching under root.
- | **Relative Path:** Use relative path in search result and generated text.
- | **File Masks:** File masks using in search to avoid useless file. Wild char supported.
- | **Case Sensitive:** Whether case sensitive when searching using file masks.
- | **Search:** Search the files according to the selected root directory and options.
- | **Generate:** If search result is not empty, pressing this button to generate the file list. If "Auto Generate" checkbox is checked, the file list will be updated automatically without clicking this button.
- | **Directories:** Whether list the directories in generated list, defaultly NOT.
- | **Files:** Whether list the files in generated list.
- | **Auto Generate:** Whether auto generate file list after tree content changed. If not checked, we need to press "Generate" button to update the file list.
- | **Save:** Save list to a text file.

Right click to popup the popupmenu. User can do some common operation to the tree contents. They do not affect the disk files, only the tree contents.

PS. The screenshots in this help document are an example of analyzing executable files of Delphi 5 and its VCL source code files. It's an good example for this tool.

## Links

[CnPack IDE Wizards License](#)

[About CnPack](#)