SNiFF+ 3

Release 3.x
for Unix and Windows

Integrating SNiFF+ with Rational Suite

TakeFive
software
An Integrated Systems Company

TakeFive Software GmbH
5020 Salzburg, Austria
E-mail: *info@takefive.co.at*

TakeFive Software, Inc
Cupertino, CA
E-Mail: *info@takefive.com*

Integration with Rational Purify, Rational Quantify, Rational Visual Pure Coverage

Rational Purify

Run-time errors, including memory leaks, are difficult errors to locate and extremely important to correct. They cause a wide variety of problems, often far from the error itself. It's not unusual for a program containing memory errors to appear to be working well—until a chance event triggers a surprise failure.

Purify is the fastest and most comprehensive run-time error detection tool available for C/C++. It detects errors both in your own code and in the components your software uses, even when you don't have the source code. Purify requires no special builds, and lets you customize error detection for each component in your program.

Purify reports memory errors such as the following, and many others, before they actually occur, so you can resolve them before they do any damage:

- Array bounds errors
- Access through dangling pointers
- Uninitialized memory reads
- Memory allocation errors
- Memory leaks
- Errors in third-party code and libraries

Use the following steps to Purify your code:

1. Instrument and run the program
2. Analyse the error view messages
3. Correct errors in source code and rerun the program

Using Purify, you can check C/C++ programs, including complex multi-threaded, multi-process applications. For example: COM-enabled applications using OLE and ActiveX controls, DLLs including Windows DLLs and Microsoft Foundation Class Library DLLs, C/C++ components embedded within Visual Basic applications, Internet Explorer, Netscape Navigator, or any Microsoft Office application. You can also check programs running in Windows CE Emulation Mode on Windows NT.

Purify also checks calls to Windows API functions, validating parameters such as memory handles and pointers. Purify can check GDI, Internet services, system registry, and COM and OLE interface APIs.

Integration goals

Functions Purify to SNiFF+

Jump from Purify defect to the source code (SNiFF+ Source Editor)

Functions SNiFF+ to Purify

Creating a custom menu with the entries:

- Start Purify
- Run `<target>` with Purify
- Open a saved datafile

Limitations

The SNiFF+ Source Editor doesn't get the focus.

Installing the Integration

Make sure that your PATH points to the directory, where purify.exe is located (<rose_install_dir>/purify).

Custom menus

To create custom menus in the Source Editor of SNiFF+ you have to make some changes in the `SiteMenus.sniff` file, which is located in the `%SNIFF_DIR%\config` folder.

> **Note**
>
> The Variable `%SNIFF_DIR%` used throughout this paper must point to your SNiFF+ installation directory

Open your `SiteMenus.sniff` file in any text specific Editor. Inserting a custom menu in SNiFF+ is very easy; you just have to insert the following words in the correct order.
For example:

`^Editor`	Specifies the location of the menu:	
	^ProjectEditor	Menu in the Project Editor
	^Editor	Menu in the Source Editor
`>Purify`	Specifies the name of the menu	

Following lines are required

```
##################################
#    SOURCE  EDITOR              #
##################################
```

```
^Editor

>Purify

shell "Start Purify" "purify.exe &"

FileNameDialog "Select executable to run with purify"
"purify /Out= %t" "Choose a file to run with purify"

FileNameDialog "Select datafile to open with purify"
"purify %t" "choose *.pfy datafile to open with purify"
```

Description of the lines

```
shell "Start Purify" "purify.exe &"
```

This line creates a menu called **Start Purify** which starts Purify.

```
FileNameDialog "Select executable to run with purify"
"purify /Out= %t" "Choose a file to run with purify"
```

This line creates a menu called **Select executable to run with purify**. By clicking on this menu a file name dialog appears. You can then choose your executable you want to run with Purify. After selecting the executable by double clicking on it or marking it and pressing <ENTER>, the executable purify.exe will be started with the parameters /Out= <filename> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.

```
FileNameDialog "Select datafile to open with purify"
"purify %t" "choose *.pfy datafile to open with purify"
```

This line creates a menu called **Select datafile to open with purify**. By clicking on this menu a file name dialog appears. You can then choose your *.pfy file (*.pfy are datafiles of Purify) you want to open with Purify. After selecting the datafile by double clicking on it or marking it and pressing <ENTER> the executable purify will be started with the parameters <filename.pfy> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.

Jumping from Purify defect to the source code (SNiFF+ Source Editor)

You must set the environment variable `SNiFF_SESSION_ID` to session1.

In Rational Purify you can specify your own Source Editor:

- Select **Settings > Preferences…**.
- Select the Source Code tab.

In the **Use the following editor** field enter the following command:

```
sniffaccessw edit_file "*/" %1 %2
```

Rational Quantify

Your customers want the fastest possible software. They want your program to work instantaneously and make the most of their computing resources. Inferior performance reduces their satisfaction with the features you worked so hard to include. So what can you do about it?

The practical solution is to identify bottlenecks, and then reduce or eliminate them, through systematic performance engineering. Begin monitoring performance just as soon as you have a working program, when it's easiest and most economical to make structural changes. Then continue tuning performance until you're ready to ship. But how can you get the data you need for performance engineering?

Visual Quantify provides a complete, accurate set of performance data for your program and its components — and provides it in an understandable and usable format so that you can see exactly where your program spends most of its time. When you use Visual Quantify to profile your code, faster code is just a few simple steps away:

1. Collect performance data
2. Analyze performance data
3. Compare performance data

Visual Quantify measures performance for:

- C/C++ code in .exes, .dlls, OLE/ActiveX controls, and COM objects
- Visual Basic projects or p-code .exes (Visual Basic 6.0), native-code .exes (Visual Basic 5.0 or later), .dlls, OLE/ActiveX controls, and COM objects
- Java applets, class files, and code launched by container programs (in conjunction with the Microsoft virtual machine for Java)
- Components launched from container programs such as Microsoft Internet Explorer, the Microsoft Transaction Server, jexegen'd executables, Jview.exe, Tstcon32.exe, Netscape Navigator, or any Microsoft Office application
- Microsoft Excel and Microsoft Word plug-ins
- Programs running in Windows CE Emulation Mode on Windows NT

Integration goals

Functions SNiFF+ to Quantify

Creating a custom menu with the entries:

- Start Quantify
- Run `<target>` with Quantify
- Open a saved datafile

Installing the Integration

Make sure that your PATH points to the directory, where quantify.exe is located (<rose_install_dir>/quantify).

Custom menus

To create custom menus in the Source Editor of SNiFF+ you have to make some changes in the `SiteMenus.sniff` file, which is located in the `%SNIFF_DIR%\config` folder. Open your `SiteMenus.sniff` file in any text specific Editor. Inserting a custom menu in SNiFF+ is very easy; you just have to insert the following words in the correct order.

For example:

`^Editor`	Specifies the location of the menu:	
	^ProjectEditor	Menu in the Project Editor
	^Editor	Menu in the Source Editor
`>Quantify`	Specifies the name of the menu	

Following lines are required

```
##################################

#    SOURCE  EDITOR              #

##################################


^Editor

>Quantify


shell "Start Quantify" "quantify.exe &"


FileNameDialog "Select executable to run with quantify"
"quantify /Out= %t" "Choose a file to run with quantify"
```

```
FileNameDialog "Select datafile to open with quantify"
"quantify %t" "choose *.qfy datafile to open with quantify"
```

Description of the lines

```
shell "Start Quantify" "quantify.exe &"
```

This line creates a menu called **Start Quantify** which starts Quantify when selecting it.

```
FileNameDialog "Select executable to run with quantify"
"quantify /Out= %t" "Choose a file to run with quantify"
```

This line creates a menu called **Select executable to run with quantify**. By clicking on this menu a file name dialog appears. You can choose then your executable you want to run with Quantify. After selecting the executable by double clicking on it or marking it and pressing <ENTER>, the executable quantify.exe will be started with the parameters /Out= <filename> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.

```
FileNameDialog "Select datafile to open with quantify"
"quantify %t" "choose *.qfy datafile to open with quantify"
```

This line creates a menu called **Select datafile to open with quantify**. By clicking on this menu a file name dialog appears. You can then choose your *.qfy file (*.qfy are datafiles of Quantify) you want to open with Quantify. After selecting the datafile by double clicking on it or marking it and pressing <ENTER> the executable quantify.exe will be started with the parameter <filename.qfy> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.

Rational Pure Coverage

Rational Pure Coverage is a tool which tests your program on never tested and used Source Code.

Integration goals

Functions SNiFF+ to Pure Coverage

Creating a custom menu with the entries:

- Start Pure Coverage
- Run `<target>` with Pure Coverage
- Open a saved datafile

Installing the Integration

Make sure that your PATH points to the directory, where coverage.exe is located (<rose_install_dir>/coverage).

Custom menus

To create custom menus in the Source Editor of SNiFF+ you have to make some changes in the `SiteMenus.sniff` file, which is located in the `%SNIFF_DIR%\config` folder. Open your `SiteMenus.sniff` file in any text specific Editor. Inserting a custom menu in SNiFF+ is very easy; you just have to insert the following words in the correct order.
For example:

`^Editor`	Specifies the location of the menu:	
	^ProjectEditor	Menu in the Project Editor
	^Editor	Menu in the Source Editor
`>”Pure Coverage”`	Specifies the name of the menu	

Following lines are required

```
##################################
#    SOURCE  EDITOR              #
##################################

^Editor
>"Pure Coverage"
shell "Start Pure Coverage" "coverage.exe &"
```

```
FileNameDialog "Select executable to run with Pure Coverage"
"coverage /Out= %t" "Choose a file to run with Pure Coverage"
```

```
FileNameDialog "Select datafile to open with Pure Coverage"
"coverage %t" "choose *.cfy datafile to open with Pure
Coverage"
```

Description of the lines

```
shell "Start Pure Coverage" "coverage.exe &"
```

This line creates a menu called **Start Pure Coverage** which starts Pure Coverage when selecting it.

```
FileNameDialog "Select executable to run with Pure Coverage"
"coverage /Out= %t" "Choose a file to run with Pure Coverage"
```

This line creates a menu called **Select executable to run with Pure Coverage**. By clicking on this menu a file name dialog appears. You can then choose your executable you want to run with Pure Coverage. After selecting the executable by double clicking on it or marking it and pressing <ENTER>, the executable coverage.exe will be started with the parameters /Out= <filename> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.

```
FileNameDialog "Select datafile to open with Pure Coverage"
"coverage %t" "choose *.cfy datafile to open with Pure
Coverage"
```

This line creates a menu called **Select datafile to open with Pure Coverage**. By clicking on this menu a file name dialog appears. You can then choose your *.cfy file (*.cfy are datafiles of Pure Coverage) you want to open with Pure Coverage. After selecting the datafile by double clicking on it or marking it and pressing <ENTER> the executable coverage.exe will be started with the parameter <filename.cfy> (%t represents the selected filename). The third parameter in inverted commas is the name of the file name dialog.